



**TECHNISCHE  
UNIVERSITÄT  
DRESDEN**

---

Fakultät Elektrotechnik und Informationstechnik

---

# **Linked Enterprise Data als semantischer, integrierter Informationsraum für die industrielle Datenhaltung**

Linked Enterprise Data as semantic and integrated  
information space for industrial data

Markus Graube

Der Fakultät Elektrotechnik und Informationstechnik  
der Technischen Universität Dresden

zur Erlangung des akademischen Grades eines

**Doktoringenieurs  
(Dr.-Ing.)**

genehmigte Dissertation

Vorsitzender: Prof. Dr. tech. Klaus Janschek

Gutachter: Prof. Dr.-Ing. Leon Urbas

Prof. Dr.-Ing. Christian Diedrich

Tag der Einreichung: 24.11.2016

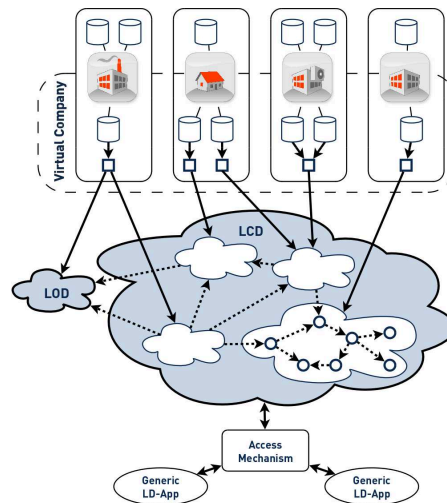
Tag der Verteidigung: 01.03.2018





## Linked Enterprise Data als semantischer, integrierter Informationsraum für die industrielle Datenhaltung

Zunehmende Vernetzung und gesteigerte Flexibilität in Planungs- und Produktionsprozessen sind die notwendigen Antworten auf die gesteigerten Anforderungen an die Industrie in Bezug auf Agilität und Einführung von Mehrwertdiensten. Dafür ist eine stärkere Digitalisierung aller Prozesse und Vernetzung mit den Informationshaushalten von Partnern notwendig. Heutige Informationssysteme sind jedoch nicht in der Lage, die Anforderungen eines solchen integrierten, verteilten Informationsraums zu erfüllen.

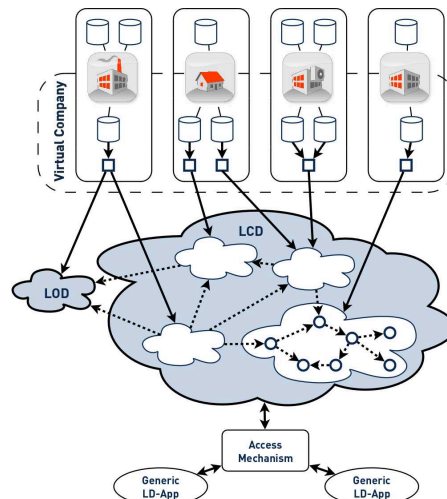


Ein vielversprechender Kandidat ist jedoch Linked Data, das aus dem Bereich des Semantic Web stammt. Aus diesem Ansatz wurde Linked Enterprise Data entwickelt, welches die Werkzeuge und Prozesse so erweitert, dass ein für die Industrie nutzbarer und flexibler Informationsraum entsteht. Kernkonzept dabei ist, dass die Informationen aus den Spezialwerkzeugen auf eine semantische Ebene gehoben, direkt auf Datenebene verknüpft und für Abfragen sicher bereitgestellt werden. Dazu kommt die Erfüllung industrieller Anforderungen durch die Bereitstellung des Revisionierungswerkzeugs R43ples, der Integration mit OPC UA über OPCUA2LD, der Anknüpfung an industrielle Systeme (z.B. an COMOS), einer Möglichkeit zur Modelltransformation mit SPARQL sowie feingranularen Informationsabsicherung eines SPARQL-Endpunkts.



## **Linked Enterprise Data as semantic and integrated information space for industrial data**

Increasing collaboration in production networks and increased flexibility in planning and production processes are responses to the increased demands on industry regarding agility and the introduction of value-added services. A solution is the digitalisation of all processes and a deeper connectivity to the information resources of partners. However, today's information systems are not able to meet the requirements of such an integrated, distributed information space.



A promising candidate is Linked Data, which comes from the Semantic Web area. Based on this approach, Linked Enterprise Data was developed, which expands the existing tools and processes. Thus, an information space can be created that is usable and flexible for the industry. The core idea is to raise information from legacy tools to a semantic level, link them directly on the data level even across organizational boundaries, and make them securely available for queries. This includes the fulfillment of industrial requirements by the provision of the revision tool R43ples, the integration with OPC UA via OPCUA2LD, the connection to industrial systems (for example to COMOS), a possibility for model transformation with SPARQL as well as fine granular information protection of a SPARQL endpoint.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.1.1	Digitale Anlage . . . . .	2
1.1.2	Semantic Web . . . . .	3
1.2	Zielstellung . . . . .	4
1.2.1	Kernthese . . . . .	4
1.2.2	Einzelthesen . . . . .	4
1.3	Lösungsansatz . . . . .	5
1.3.1	Einordnung und Abgrenzung der Arbeit . . . . .	5
1.3.2	Aufbau der Arbeit . . . . .	7
<b>2</b>	<b>Grundlagen und Stand der Technik</b>	<b>9</b>
2.1	Industrie 4.0 und CPS . . . . .	9
2.2	Informationsraum . . . . .	11
2.2.1	Begriffsdefinition . . . . .	11
2.2.2	Data Warehouse . . . . .	12
2.2.3	Middleware . . . . .	13
2.3	Informationsmodellierung . . . . .	14
2.3.1	Semantik vs. Syntax . . . . .	14
2.3.2	Informationsmodell . . . . .	15
2.3.3	Ontologie . . . . .	15
2.4	Modelltransformation . . . . .	16
2.4.1	Grundlegendes Konzept . . . . .	16
2.4.2	Graph-Transformationen . . . . .	16
2.4.2.1	Graph-Transformationsansätze . . . . .	17
2.4.2.2	Triple Graph Grammars . . . . .	18
2.4.2.3	Anwendungsfälle für Triple Graph Grammars . . . . .	18
<b>3</b>	<b>Anforderungsanalyse</b>	<b>21</b>
3.1	Vorgehen zur Anforderungsermittlung . . . . .	21
3.2	Anforderungen an industrielle Informationsräume . . . . .	22
3.2.1	Beschreibungsmittel . . . . .	22
3.2.1.1	Mächtige Informationsmodellierung (B1) . . . . .	23

3.2.1.2	Semantik (B2)	23
3.2.1.3	Erweiterbarkeit (B3)	24
3.2.1.4	Selbstbeschreibungsfähigkeit (B4)	24
3.2.1.5	Organisationsübergreifende Verknüpfungen (B5)	25
3.2.2	Methoden	26
3.2.2.1	Abstraktion der Kommunikation (M1)	26
3.2.2.2	Statische Daten (M2)	26
3.2.2.3	Dynamische Daten (M3)	27
3.2.2.4	Informationssicherheit (M4)	27
3.2.2.5	Dienstorientierung (M5)	28
3.2.3	Werkzeuge	28
3.2.3.1	Modelltransformationen (W1)	28
3.2.3.2	Revisionierung (W2)	29
3.2.3.3	Werkzeugunterstützung für die Entwicklung (W3)	30
3.2.3.4	Anbindung vorhandener Daten (W4)	30
3.2.3.5	Unterstützung zur Nachverfolgbarkeit/Inspi- zierbarkeit zur Laufzeit (W5)	30
<b>4</b>	<b>Analyse bestehender Ansätze</b>	<b>33</b>
4.1	Semantic Web	33
4.1.1	Ziele	33
4.1.2	Prinzipien	35
4.1.3	Best Practises	35
4.2	ISO 15926	36
4.3	OPC UA	37
4.4	Eclipse EMF	38
4.5	AutomationML	39
4.6	WBEM/CIM	40
4.7	Zusammenfassende Bewertung	41
<b>5</b>	<b>Konzept - Linked Enterprise Data (LED)</b>	<b>45</b>
5.1	Abdeckung der Anforderungen durch Linked Data	45
5.1.1	Erfüllte Anforderungen	45
5.1.2	Notwendige Erweiterung zu LED für einen industriellen Informationsraum	47
5.2	Kernkonzept von Linked Enterprise Data	48
5.3	Dezentrale Architektur	51
5.3.1	LED-Architektur	51



5.3.2	Verteilte Abfragen (Federated Queries) . . . . .	52
5.4	Informationsbereitstellung - LD Adapter . . . . .	54
5.4.1	Dynamische Daten . . . . .	54
5.4.1.1	Linked Data Plattform (LDP) . . . . .	55
5.4.1.2	OPC UA Adapter . . . . .	55
5.4.2	Quasi-Statische Daten . . . . .	57
5.4.2.1	XIWrap . . . . .	57
5.4.2.2	D2RQ . . . . .	57
5.4.2.3	Legacy-Daten . . . . .	58
5.5	Zugriffssicherung für Linked Enterprise Data . . . . .	58
5.6	Versionierung . . . . .	60
5.7	Modell-Transformation . . . . .	61
5.7.1	Graph-Transformation . . . . .	62
5.7.1.1	Abbildung mathematischer Graphen auf Linked Data . . . . .	62
5.7.1.2	Werkzeuge . . . . .	62
5.7.1.3	Graph-Transformation für Linked Data . . . . .	64
5.7.2	Modell-Synchronisierung . . . . .	65
5.8	Prozesse für LED . . . . .	66
5.8.1	Standardisierung vs. Flexibilität mit LED . . . . .	66
5.8.2	Einordnung von LED in die Geschäftsprozesse von Virtuellen Unternehmen . . . . .	67
<b>6</b>	<b>Implementierung - Notwendige Komponenten für LED</b>	<b>69</b>
6.1	ComVantage-Architektur . . . . .	70
6.2	OPCUA2LD . . . . .	71
6.2.1	Mapping zwischen OPC UA und Linked Data . . . . .	72
6.2.2	Zugriffsmethoden . . . . .	73
6.2.2.1	Browsing und Attribute lesen . . . . .	73
6.2.2.2	Zugriff auf historische Daten . . . . .	74
6.2.2.3	Werte schreiben . . . . .	75
6.2.2.4	Methoden aufrufen . . . . .	76
6.2.2.5	Nicht unterstützte OPC UA Dienste . . . . .	76
6.2.3	Implementierung . . . . .	77
6.3	CESS - COMOS Enterprise Server Service . . . . .	77
6.4	SPARQL Security Rewriter . . . . .	79
6.5	R43ples . . . . .	85
6.5.1	Konzept und Architektur . . . . .	85
6.5.2	R43ples Interface . . . . .	86

6.5.3	Merging-Funktionalitäten . . . . .	87
6.5.4	Leistungsmetriken . . . . .	87
6.5.4.1	Speicherbedarf . . . . .	87
6.5.4.2	Antwortzeiten für die Wiederherstellung alter Revisionen . . . . .	87
6.6	SMT (SPARQL Model Transformation) . . . . .	88
6.7	RDF UML Diagramm . . . . .	90
6.7.1	Reine RDF Visualisierung . . . . .	91
6.7.2	Objektdiagramm . . . . .	91
6.7.3	Klassendiagramm . . . . .	92
6.7.4	Umsetzung . . . . .	93
<b>7</b>	<b>Kritische Reflexion</b>	<b>95</b>
7.1	Bewertung der Methodik . . . . .	95
7.2	Bewertung der Ergebnisse . . . . .	96
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>99</b>
8.1	Zusammenfassung . . . . .	99
8.2	Innovationen und Grenzen . . . . .	99
8.3	Ausblick . . . . .	101
<b>Anhang A</b>	<b>Kriterien für die Anforderungsbewertung von Informa- tionsräumen</b>	<b>107</b>
<b>Anhang B</b>	<b>Abbildungen und Listings</b>	<b>111</b>
<b>Anhang C</b>	<b>Linked Data Grundlagen</b>	<b>125</b>
C.1	Technologien . . . . .	125
C.1.1	IRI/URI . . . . .	125
C.1.2	RDF . . . . .	125
C.1.3	HTTP . . . . .	127
C.1.4	SPARQL . . . . .	127
C.1.5	Links . . . . .	127
C.1.6	Ontologien . . . . .	127
C.2	Werkzeuge . . . . .	128
C.2.1	Frameworks . . . . .	128
C.2.2	TripleStores . . . . .	128
C.2.3	Ontologie-Management . . . . .	130

C.2.4	Werkzeuge zur Verknüpfung von Datensätzen . . . .	130
-------	---	-----

<b>Literaturverzeichnis</b>		<b>133</b>
-----------------------------	--	------------



## Abbildungsverzeichnis

1.1	Integration von Automatisierungssystemen in unterschiedlichen Ebenen [The+09] . . . . .	3
1.2	Linked Enterprise Data zur Lösung von Interoperabilitätsproblemen . . . . .	6
1.3	Gliederung der Dissertation . . . . .	8
2.1	Virtuelles Unternehmen . . . . .	10
2.2	Modelltransformation [CH06] . . . . .	17
2.3	Deklarative TGG Regel [KS06] . . . . .	18
5.1	Semantic Web Stack [zaz11b] . . . . .	46
5.2	Konzept in einer Nusschale . . . . .	50
5.3	Konzept für den Zugriff auf dynamische Daten . . . . .	56
5.4	SAPRQL security [Ort+13] . . . . .	59
5.5	Graphtransformationsregel zur Ersetzung des Prädikats <i>rdfs:label</i> mit <i>dc:title</i> . . . . .	64
5.6	Linked Enterprise Data (orange) im Gesamtprozess eines virtuellen Unternehmens nach [Zie+14] . . . . .	68
6.1	Überblick über die LED-Architektur . . . . .	71
6.2	OPCUA2LD Adapter Konzept . . . . .	72
6.3	CESS Sequenzdiagramm für den Export von RDF . . . . .	79
6.4	Architektur von R43ples als SPARQL-Proxy . . . . .	85
6.5	Konzept für SPARQL Transformation . . . . .	89
6.6	RDF-UML-Objektdiagramm . . . . .	92
6.7	RDF-UML-Klassendiagramm . . . . .	93
B.1	Abbildung der implementierten Komponenten auf Architektur	113
B.2	Zusammenhang zwischen mathematischen Graphen und Linked Data . . . . .	115
B.3	Ableitung von operationalen Regeln aus Abb. 2.3 [KS06] . .	116
B.4	CESS Verteilungsdiagramm . . . . .	117
B.5	Graphische RDF Repräsentation des Beispiel-Datensatzes aus Listing B.1 . . . . .	118

B.6	Großes RDF-UML-Objektdiagramm mit 73 Tripeln . . . . .	119
B.7	Informationsmodell eines einfachen Revisionsgraphen in R43ples von dem Graphen <i>http://test</i> mit zwei Revisionen und den dazugehörigen Commit . . . . .	120
B.8	Screenshot von R43ples . . . . .	121
B.9	Vergleich der Antwortzeiten von R43ples zwischen der Query Rewriting Option (dunkelgrau) und der Option mit Erstellung eines temporären Graphen (hellgrau) für eine ansteigende Entfernung zur herzustellenden Revision (links), der Größe des Changesets (Mitte) und der Datensatzgröße (rechts) . . . . .	122
B.10	Konzept für Query-Transformation . . . . .	123
C.1	Semantic Web Stack [zaz11b] . . . . .	126

## Tabellenverzeichnis

4.1	Bewertung der Technologien hinsichtlich der aufgestellten Anforderungen . . . . .	42
6.1	Abbildung der LED-Konzepte auf Laufzeitkomponenten . . .	69
6.2	Ergebnis zur Abfrage aus Listing 6.11 ohne Informationsabsicherung . . . . .	81
6.3	Ergebnis zur Abfrage aus Listing 6.15 mit einer Einschränkung auf die sichtbaren Views von <i>user:operator23</i> . . . . .	81
A.1	Kriterien hinsichtlich der Erfüllung der Anforderungen im Bereich Beschreibungsmittel . . . . .	108
A.2	Kriterien hinsichtlich Anforderungsbewertung im Bereich Methoden . . . . .	109
A.3	Kriterien hinsichtlich Anforderungsbewertung im Bereich Werkzeuge . . . . .	110





## Quelltextverzeichnis

5.1	Federated SPARQL Query . . . . .	53
5.2	R43ples Select Query . . . . .	61
5.3	SPARQL Repräsentierung der Transformationsregel aus Ab- bildung 5.5 . . . . .	65
6.1	GET Request für das Browsing eines Knotens . . . . .	73
6.2	Ergebnis für Anfrage aus Listing 6.1 . . . . .	73
6.3	Ergebnis eines GET Requests zum Auslesen einer OPC UA Variable . . . . .	74
6.4	Ergebnis für Anfrage aus Listing 6.3 . . . . .	74
6.5	Ergebnis eines GET Requests für den Zugriff auf historische Daten . . . . .	75
6.6	Ergebnis für Anfrage aus Listing 6.5 . . . . .	75
6.7	POST Request zum Schreiben einer OPC UA Variable . . . . .	75
6.8	Ergebnis zu Listing 6.7 . . . . .	76
6.9	POST Requests für den Aufruf einer OPC UA Methode . . . . .	76
6.10	Ergebnis für den Aufruf der Methode aus Listing 6.9 . . . . .	76
6.11	Query zur Abfrage der Bekannten von <i>ins:jsmith</i> mit Namen und optionaler Mail-Adresse . . . . .	80
6.12	Generierung des Views <i>view:operator1</i> . . . . .	82
6.13	Generierung des Views <i>view:manager</i> . . . . .	83
6.14	Zuweisung von Nutzern zu Views . . . . .	83
6.15	Umgeschriebene Query aus Listing 6.11 für den Nutzer <i>user:operator23</i> . . . . .	84
B.1	Beispielhafter RDF Datensatz in Turtle-Serialisierung mit 19 Triples . . . . .	112
B.2	Informationsmodell zu Listing 6.11 . . . . .	114



## Abkürzungsverzeichnis

CAE	Computer Aided Engineering)
CESS	Comos Enterprise Server Service
CIM	Common Information Model
CIM	Computer-Integrated Manufacturing
COLLADA	COLLABorative Design Activity
CPPS	Cyber Physical Production System
CPS	Cyber Physical System
CWA	Closed World Assumption
DEXPI	Data Exchange for the Process Industry
DMTF	Distributed Management Task Force
DWH	Data Warehouse
ERP	Enterprise Resource Planning
EUPL	European Public License
GPL	GNU Public License
HTTP	Hypertext Transfer Protocol
I4.0	Industrie 4.0
JSON	JavaScript Object Notation
LCD	Linked Confidential Data
LD	Linked Data
LDP	Linked Data Platform

LED	Linked Enterprise Data
LOD	Linked Open Data
MES	Manufacturing Execution System
MMS	Mensch-Maschine-Schnittstelle
MOF	Managed Object Format
MSO	Maintenance Structure Ontology
NAMUR	Normen-Arbeitsgemeinschaft Mess- und Regeltechnik
OPC	OLE for Process Control
OPC UA	OPC Unified Automation
OPEX	Operational expenditure
OWA	Open World Assumption
OWL	Web Ontology Language
PLS	Prozessleitsystem
R43ples	Revision for Triples
RAMI	Referenzarchitektur-Modell Industrie 4.0
RDF	Resource Description Framework
RDFS	RDF Schema
REST	Representational State Transfer
RFC	Request for Comments
RFID	Radio Frequency Identification
RMO	Revision Management Ontology
SLA	Service Level Agreement
SMT	SPARQL Modell Transformation

SoS	System of Systems
SPARQL	SPARQL Protocol And RDF Query Language
SPIN	SPARQL Inference Notation
SQT	SPARQL Query Transformation
TGG	Triple Graph Grammars
UML	Unified Modeling Language
URI	Uniform Resource Identifier
VCS	Version Control System
VRML	Virtual Reality Modeling Language
WBEM	Web-Based Enterprise Management
XML	Extensible Markup Language



# 1 Einleitung

## Aufbau und Einordnung des Kapitels

Im ersten Kapitel werden die Ansätze zur digitalen Anlage und zum Semantic Web beschrieben und aus der möglichen Zusammenführung die Dissertation motiviert. Darauf aufbauend wird die Zielstellung der Arbeit mit Thesen konkretisiert und der Lösungsansatz beschrieben, mit der die Thesen untersucht werden sollen.

## 1.1 Motivation

Die gesamte Produktionslandschaft und somit auch die Automatisierungstechnik unterliegen momentan einem starken Wandel in Richtung Digitalisierung und Vernetzung von Dienstleistungen. Dies passiert im Zuge der aktuellen Entwicklung zu Industrie 4.0 und Cyber-Physical-Systems (CPS), mit denen neuartige Produktionsbeziehungen und Wechselwirkungen einhergehen, um auf den zunehmend volatilen Märkten Flexibilitäts- und Geschwindigkeitsvorteile aufrecht zu gewinnen und zu erhalten.

Somit werden in Zukunft häufiger verschiedene Organisationen Teile ihrer Daten in abgegrenzten Anwendungsgebieten teilen, in denen sie eine *virtuelle Fabrik* bilden und gemeinsam als Unternehmen auftreten und Produkte oder Dienstleistungen verkaufen. Damit gehen die Organisationen eine unternehmensübergreifende Kollaboration ein. Die Fähigkeit, schnell neue Partner in den Wertschöpfungsprozess zu integrieren, stellt eine Kernherausforderung dar und sollte durch den Einsatz von leichtgewichtigen Ansätzen zur Datenintegration und Werkzeugintegration unterstützt werden. Die dafür notwendigen Formate und Schnittstellen müssen dabei aber ebenso auf Veränderbarkeit und Erweiterbarkeit ausgelegt sein. Zudem besteht eine Herausforderung darin, die Technologie für das Teilen von Daten dezentral zu installieren und zu nutzen. Denn eine zentral verwaltete Plattform kann den Eigeninteressen der anderen beteiligten Unternehmen zum Schutz der eigenen Daten zuwider laufen.

Diese Entwicklung geht mit der Auflösung der klassischen Automatisierungspyramide einher [SMS11]. Die notwendigen Funktionen bleiben zwar

weiterhin hierarchisch geordnet, die feste Zuordnung dieser Funktionen zu den Geräten, auf denen diese laufen, schwindet jedoch zunehmend, so dass hier eine größere Flexibilität Einzug hält.

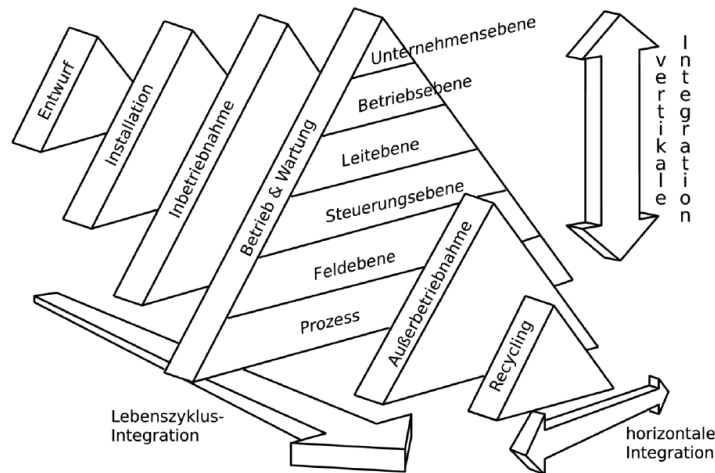
### 1.1.1 Digitale Anlage

Jede real existierende Anlage erhält zunehmend eine Abbildung in der digitalen Welt. Diese wird in der ersten Phasen der Anlagenplanung initiiert und aufeinander aufbauend durch verschiedene Gewerke verfeinert und um weitere (Teil-)Modelle erweitert. Aber auch während der Betriebsphase muss sich die digitale Anlage fortwährend den Veränderungen der realen Anlage anpassen. Damit bleiben die in der digitalen Anlage vorhandenen Simulationsmodelle gültig und ermöglichen so eine Optimierung der Betriebsführung und Wartungsaktivitäten. Eine Hauptherausforderung für eine konsistente digitale Anlage stellen dabei die Umbrüche zwischen beteiligten Gewerken und einzelnen Phasen im Lebenszyklus der Anlage dar.

Die Bedeutsamkeit des Themas kann an der steigenden Komplexität von Anlagen bei der gleichzeitigen sinkenden zur Verfügung stehenden Zeit für die Planung dargestellt werden. Ein typisches Mengengerüst einer Chemieanlage mit einem Investitionsvolumen von 500 Millionen € kann mehrere 10.000 Feldgeräte enthalten [Kir07; SWH09], die wiederum ein Vielzahl an Eingängen und Ausgängen enthalten können. Dabei wird eine Verkürzung der Entwicklungszeit von 10 Jahren auf 5 Jahre angestrebt. Dieses soll durch eine stärkere Teilung und Parallelisierung der Arbeitsaufgaben erreicht werden. Diese Komplexität kann nur durch eine geschickte Integration aller Aspekte der digitalen Anlage gelöst werden, aus der aufbereitete Informationen abgeleitet werden können. Mit Hilfe dieser lassen sich leichter Entscheidungen für eine zügige und sichere Planung und für einen wirtschaftlichen Betrieb von Anlagen treffen.

Die Integration in der digitalen Anlage umfasst verschiedene Ebenen. So muss einerseits horizontal und vertikal zur Automatisierungspyramide integriert werden, andererseits wird die Vernetzung von Information entlang des Lebenszyklus der Anlagen, Produkte, Automatisierungsgeräte und Automatisierungssysteme immer wichtiger [The+09] (siehe auch Abbildung 1.1). Daher muss ein zukünftiger Informationsraum sowohl auf Änderungen innerhalb der Zeit reagieren können als auch auf Vernetzung zu anderen Systemen ausgelegt sein.





**Abbildung 1.1:** Integration von Automatisierungssystemen in unterschiedlichen Ebenen [The+09]

### 1.1.2 Semantic Web

Das Semantic Web ist ein großer Treiber aus der IT, der eine Vielzahl an Applikationen auf lange Sicht ändern wird. Ziel der Entwicklung ist, dass die Informationen im Web zunehmend maschinell auswertbar werden und aus Kombination von unterschiedlichen Datensätze neue Informationen erschlossen werden können, die die Beantwortung komplexer Fragestellungen erlauben. Dazu wurde eine Reihe von Basistechnologien ausgewählt und entwickelt, die lose gekoppelt zusammen ein mächtiges semantisches Informationsnetzwerk aufspannen können.

Bereits seit einiger Zeit werden Aspekte dieses Ansatzes im wissenschaftlichen Umfeld der Automatisierungstechnik diskutiert [Str+11; Los+11]. Eine umfassende Betrachtung des Konzepts für die Automatisierungstechnik steht aber immer noch aus. Auch in der Industrie erweckt dieser Ansatz mit Linked Data mittlerweile Interesse, so dass es diese Thematik in die Normungsroadmap für Industrie 4.0 [DD15] geschafft hat, ohne jedoch die Schwachstellen und Mängel dieses Ansatzes für die industrielle Nutzung aufzeigen zu können.

### 1.2 Zielstellung

Das Ziel der Arbeit besteht in der systematischen Konzeption und Umsetzung eines flexiblen und sicheren Informationsraums für die industrielle Datenhaltung auf der Basis einer ausführlichen Analyse der Anforderungen aus Theorie und Praxis. Die Arbeit soll den Nachweis erbringen, dass die Konzepte des Semantic Web für die Integrationsbemühungen der digitalen Anlage einen geeigneten Ansatz darstellen. Darüber hinaus ermöglicht der Ansatz eine flexible Weiterentwicklung der Informationsmodelle und kann sich so den sich ändernden Anforderungen der realen Welt anpassen.

#### 1.2.1 Kernthese

Die wissenschaftliche Kernthese dieser Arbeit lässt sich folgendermaßen zusammenfassen:

*Linked Enterprise Data erweitert Linked Data um Methoden und Werkzeuge und ermöglicht so eine flexible und sichere Integration von heterogenen Informationsquellen über Unternehmensgrenzen und Lebenszyklusphasen hinweg.*

#### 1.2.2 Einzelthesen

Dazu wird die These in folgende wissenschaftliche Forschungthesen aufgeteilt:

**These 1.** *Informationssysteme zur Erfüllung von Szenarien für Industrie 4.0, CPS bzw. Advanced Manufacturing stellen neue Anforderungen an gemeinsame Informationsräume.*

**These 2.** *Die Anforderungen für solche Informationsräume lassen sich in die Dimensionen Beschreibungsmittel, Methoden und Werkzeuge auftrennen.*

**These 3.** *Anforderungen an Informationsräume können von heutigen Systemen nicht vollständig erfüllt werden.*

**These 4.** *Linked Data erfüllt einen Großteil der gestellten Anforderungen, ist aber mit den heutigen Konzepten und Technologien industriell nicht einsetzbar.*

**These 5.** *Geeignete Erweiterungen im Bereich der Beschreibungsmittel, Methoden und Werkzeuge lassen daraus Linked Enterprise Data entstehen, das die erfassten Anforderungen abdeckt.*

## 1.3 Lösungsansatz

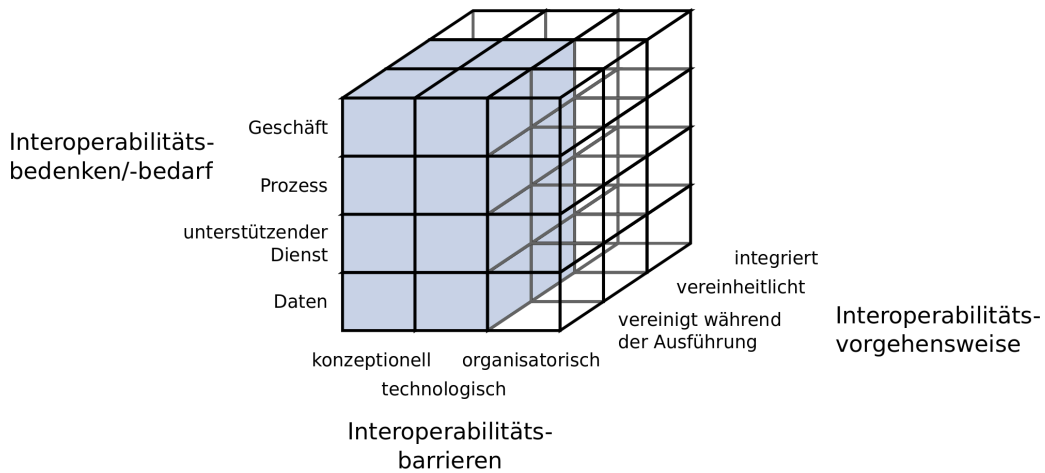
### 1.3.1 Einordnung und Abgrenzung der Arbeit

Diese Arbeit beschäftigt sich mit der Informationsintegration für industrielle Anwendung und nutzt dafür als Beschreibungsmittel Sprachelemente des Semantic Web (RDF, OWL, SPIN) sowie in der Industrie etablierte Standards. Methodisch werden Linked Data Ansätze sowie Modelltransformationen mit speziellem Fokus auf Graph Grammars verwendet. Verwendete und unterstützte Werkzeuge beziehen sich zum einen aus dem Bereich der Planungs- und Betriebswerkzeuge der Verfahrens- und Automatisierungstechnik (COMOS, OPC, PCS S7), zum anderen aus Tools des Semantic Web (Triple Stores, SPARQL, Protégé) sowie aus selbstentwickelten Werkzeugen zur Überbrückung dieser Welten (Adapter, Revisionsverwaltungssysteme).

Die Lösung soll über den gesamten Lebenszyklus von Produktionsanlagen eingesetzt werden und phasen- und gewerkeübergreifende Kollaborationen in allen Planungs- und Betriebsprozessen unterstützen. Als Ressourcen aus der Automatisierungstechnik werden dazu die digitale Anlage (aus Engineering Phase), Live Daten (aus PLS, MES, ERP), Identifikationssysteme (RFID, QR-Code) sowie mobile Informationssysteme (MIS) genutzt.

Dabei soll die Lösung nicht nur in einem Einzelunternehmen eingesetzt werden, sondern vor allem so in Organisationsverbünden eingesetzt werden, dass die *Unternehmensinteroperabilität* erhöht wird. Diese ist in DIN EN ISO 11354-1 [DCI09] als “Fähigkeit von in Unternehmungen kooperierenden Unternehmen miteinander zu kommunizieren und effektiv interagieren zu können” definiert. Dabei besteht das Bedürfnis, Interoperabilität in Daten, unterstützenden Diensten, Prozessen und Geschäften zu erreichen. Dem entgegen stehen konzeptionelle, technologische und organisatorische Barrieren, ausgelöst durch Inkompatibilitäten und Fehlanpassungen. Diese können durch eine integrierte, eine vereinheitlichte und eine während der Ausführung vereinigte Vorgehensweise beseitigt werden. DIN EN ISO 11354-1 unterscheidet bei der Betrachtung von Interoperabilität drei Kategorien:

- **Interoperabilitätsbedarf und -bedenken** beschreiben die Bereiche, in denen noch Hindernisse für eine Interoperabilität bestehen, deren Beseitigung jedoch eine wichtige Lücke schließen würde. Die Norm unterscheidet hier zwischen Geschäft, Prozess, unterstützenden Diensten und Daten.
- **Interoperabilitätsbarrieren** beschreiben, ob die auftretenden Hin-



**Abbildung 1.2:** Linked Enterprise Data zur Lösung von Interoperabilitätsproblemen

dernisse für eine umfassende Interoperabilität konzeptioneller, technologischer oder organisatorischer Herkunft sind.

- **Interoperabilitätsvorgehensweise** beschreibt die Vorgehensweise zur Beseitigung der Interoperabilitätsbarrieren für die erkannten Interoperabilitätsbedarfe. Dafür gibt es drei Ansätze: integriert, vereinheitlicht oder vereinigt während der Ausführung.

Diese Arbeit versucht, mit dem hier eingeführten Konzept *Linked Enterprise Data* die Interoperabilität in einer Untermenge dieser Kategorien zu verbessern, wie in Abbildung 1.2 dargestellt. Dabei können allen Interoperabilitätsbedenken die konzeptionellen und technologischen Barrieren durch eine teils vereinheitlichte und teils erst in der Laufzeit vereinigten Vorgehensweise genommen werden.

Die Arbeit konzentriert sich somit auf die technischen Herausforderungen und Probleme, die mit einem integrierten Informationsraum verbunden sind. Organisatorische Aspekte werden nur im Zusammenhang mit technischen Fragestellungen behandelt und rechtliche Aspekte werden komplett ausgeblendet, wohlwissend dass auch diese erst eine Unternehmensinteroperabilität ermöglichen. Um die Bereitschaft und Motivation der Unternehmen zu erzeugen und steigern, solche Kooperationen einzugehen, ist jedoch die Demonstration der technischen Machbarkeit ein sinnvoller erster Schritt. Zu diesem will diese Arbeit einen Beitrag leisten. Die integrierte Vorgehensweise wird hier ausgeschlossen, da diese voraussetzt, dass sich Unternehmen bereits auf eine gemeinsame technische Basis geeinigt haben. Dafür gibt es jedoch bereits

genügend Lösungen, so dass dies keine wissenschaftliche Herausforderung mehr darstellt.

### 1.3.2 Aufbau der Arbeit

Um die vorgestellten Thesen nachvollziehbar untersuchen zu können, gliedert sich die Dissertation in folgende Kapitel, die in Abbildung 1.3 illustriert sind.

Kapitel 2 beschreibt die Grundlagen der verwendeten Technologien und den aktuellen Stand der Wissenschaft und Technik zur Integration von Informationen im industriellen Kontext. Eine Literaturrecherche zu aktuellen Entwicklungen führt zu der Anforderungsanalyse in Kapitel 3, das notwendige Voraussetzungen für den Aufbau eines integrierenden Informationsraums beschreibt. Diese Anforderungen werden in Kapitel 4 mit momentan verfügbaren Technologien abgeglichen und geeignete Kandidaten für eine Weiterentwicklung identifiziert. In Kapitel 5 wird aus dem Semantic Web und dessen Anforderungserfüllung die Grundidee und die Konzepte von *Linked Enterprise Data* (LED) abgeleitet sowie begründet, warum damit die aufgenommen Anforderungen für industrielle Informationsräume erfüllt werden können. Das Kapitel 6 beschreibt, wie das Konzept für konkrete Anwendungsfälle umgesetzt werden kann und welche Anforderungen damit erfüllt werden können. Eine Diskussion des Konzeptes und der Implementierung erfolgt in Kapitel 7. Die Arbeit schließt mit einer Zusammenfassung der Ergebnisse und einem Ausblick auf weitere Forschungsfragen in Kapitel 8.

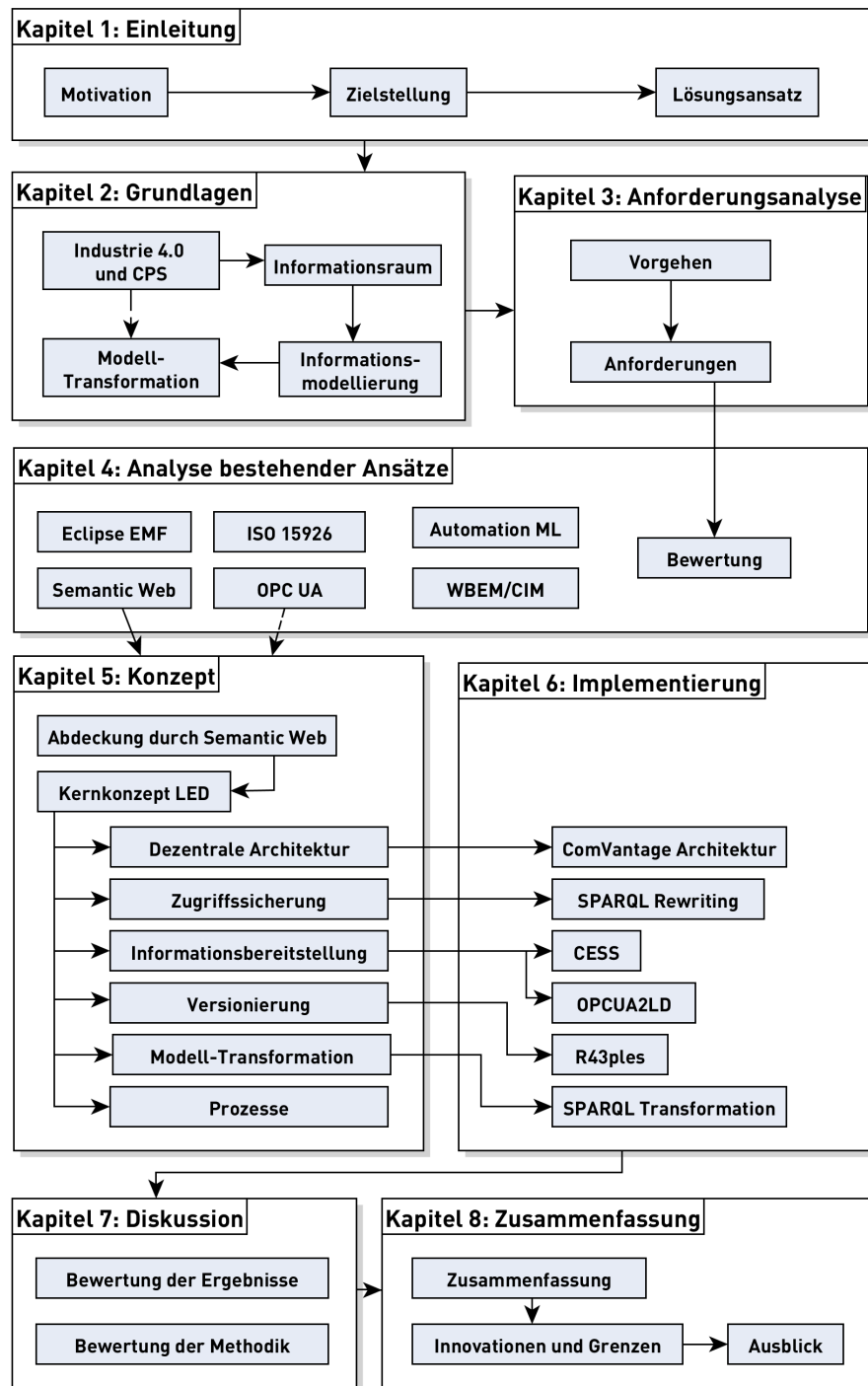


Abbildung 1.3: Gliederung der Dissertation

## 2 Grundlagen und Stand der Technik

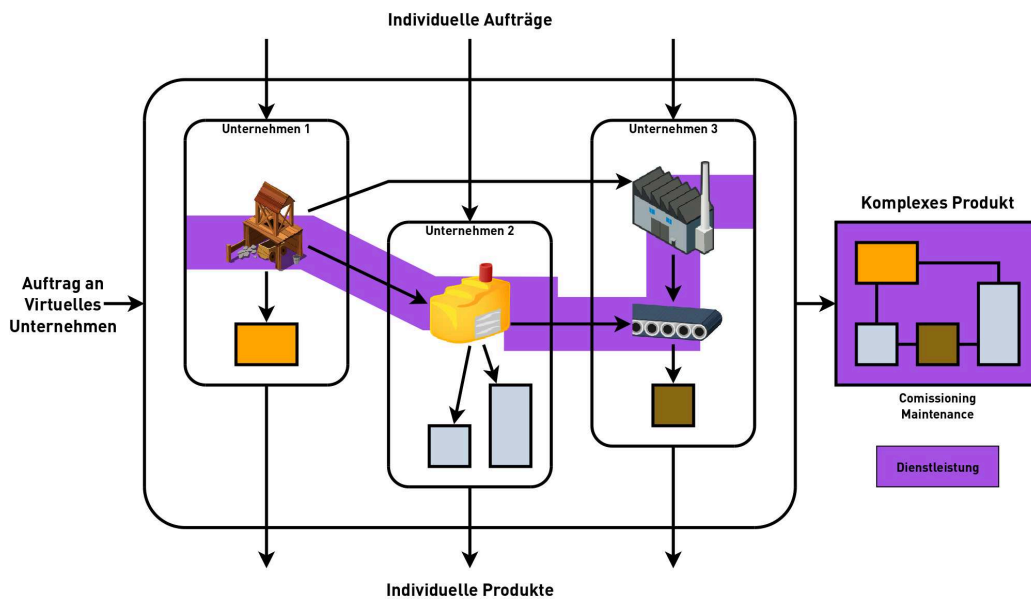
### Aufbau und Einordnung des Kapitels

In diesem Kapitel werden die grundlegenden Begriffe und Zusammenhänge des Themengebiets erklärt und anschließend der aktuelle Stand der Forschung und Technik auf dem Gebiet der Informationsintegration in der Automatisierungstechnik dargelegt. Zunächst wird dabei auf die aktuellen Trends Industrie 4.0 und Cyber-Physical Systems (CPS) eingegangen. Daraus leitet sich der Bedarf für einen gemeinsamen Informationsraum ab, bevor auf die damit zusammenhängenden Themen Informationsmodellierung und Modelltransformation eingegangen wird.

### 2.1 Industrie 4.0 und CPS

Industrie 4.0 als Initiative der deutschen Bundesregierung steht für die vierte industrielle Revolution, die durch eine verbesserte Kommunikation die Organisation und Steuerung der gesamten Wertschöpfungskette über den Lebenszyklus von Produkten verbessern soll [Pla15]. Dabei wird sich zunehmend an individualisierten Kundenwünschen orientiert. Basis ist die Verfügbarkeit aller relevanten Informationen in Echtzeit durch Vernetzung aller an der Wertschöpfung beteiligten Instanzen, um daraus einen optimalen Wertschöpfungsfluss abzuleiten. Durch die Verbindung von Menschen, Objekten und Systemen entstehen dynamische, echtzeitoptimierte und selbst organisierende, unternehmensübergreifende Wertschöpfungsnetzwerke.

Eng verbunden mit Industrie 4.0 ist der Begriff von cyber-physischem Systemen (CPS). Diese umfassen laut [Fa13] „eingebettete Systeme, Produktions-, Logistik-, Engineering-, Koordinations- und Managementprozesse sowie Internetdienste, die mittels Sensoren unmittelbar physikalische Daten erfassen und mittels Aktoren auf physikalische Vorgänge einwirken, mittels digitaler Netze untereinander verbunden sind, weltweit verfügbare Daten und Dienste nutzen und über multimodale Mensch-Maschine-Schnittstellen verfügen“. Der Hauptunterschied zu bisherigen Automatisierungssystemen ist dabei der Fakt, dass über offene, globale Informationsplattformen kommuniziert werden soll [VDI13]. Aktuelle Diskussionen zeigen, dass dabei der Mensch eine entschei-



**Abbildung 2.1:** Virtuelles Unternehmen

dende Rolle innehaben soll, um die Flexibilität in die gewünschte Richtung steuern zu können. Dabei werden große Potentiale von solchen Systemen in dem geringeren Aufwand zur Einrichtung, Pflege und Wartung sowie in einer besseren Qualität der Dokumentation gesehen [Spa+13, S. 63].

Virtuelle Unternehmen (siehe Abb. 2.1) sind ein projektbasierter Zusammenschluss von einzelnen Unternehmen, die zusammen ein komplexes Produkt und/oder Dienstleistung anbieten [Bar+94]. Dabei können diese Partnerschaften volatil und dynamisch sein, in der jeder Partner eine gewisse Expertise und Fähigkeiten einbringt. Dafür ist es jedoch notwendig, dass die Partner untereinander die jeweils benötigten Informationsentitäten teilen. Diese stellen wiederum nur einen explizit ausgewählten und freigegebenen Teil der internen Informationsmenge dar.

In diesem Zug wird Interoperabilität eine immer wichtige Eigenschaft von Informationssystemen. So sehen Drath, Fay und Barth „Interoperabilität [...] als Schlüssel zur Effizienz“ [DFB11]. Dabei sieht die VDI/VDE 2657-1 [VDI11b] Interoperabilität als „Fähigkeit unabhängiger Systeme, zusammenzuarbeiten und Informationen auszutauschen“. Dahingegen hat [DFB11] einen klaren Fokus auf Interoperabilität zwischen Engineeringwerkzeugen, die das Ziel verfolgt, „Konsistenz zwischen den Daten einer Werkzeugkette computergestützt, systematisch und wiederholt herstellen zu können“. Bei beiden



Betrachtungsweisen ist dafür jedoch die Einhaltung gemeinsamer Standards erforderlich, die beispielsweise Dateiformate oder Protokolle definieren.

Im Rahmen dieser Arbeit werden zwei Systeme kompatibel genannt, wenn diese zusammenarbeiten können. Dabei können die Systeme auf der gleichen Ebene der Automatisierungspyramide stehen oder auf unterschiedlichen. Durch das Aufbrechen der Automatisierungspyramide wird dabei der Austausch zwischen verschiedenen Ebenen immer wichtiger.

Die Aussagen der VDI/VDE 2657-1 [VDI11b] treffen nicht nur auf Middlewares zu, sondern allgemein auf Informationssysteme. So könne einem System alleine nicht die Eigenschaft der Interoperabilität zugewiesen werden, da Interoperabilität eine Eigenschaft zwischen zwei Systemen definiert. Dies sei lediglich im Kontext mit anderen Systemen möglich. Der Einsatz von Standards könne allerdings bereits Hinweise auf die Interoperabilität eines Informationssystems mit anderen Informationssystemen geben.

Eine besondere Herausforderung für Industrie 4.0 besteht darin, dass die Vorgehensweise in der Informationstechnik, in der Informatik, in der Mechanik, in der Automatisierungstechnik und in ihren Anwendungsgebieten wie Produktionstechnik, Prozesstechnik, Energietechnik und Fahrzeugbau verschieden sind und unterschiedlich lange Lebenszyklen aufweisen [VDI13]. Dieses müssen zukünftige CPS zusammenbringen.

## 2.2 Informationsraum

### 2.2.1 Begriffsdefinition

Ein zentraler Begriff dieser Arbeit ist der *Informationsraum*. Dieser soll im Folgendem erläutert werden. Ausgehend von der Definition von Nemby [New96], der einen Informationsraum als „a set of concepts and relations among them held by an information system“ sieht, haben sich eine Reihe unterschiedlicher Weiterentwicklungen ergeben.

Boisot et. al. [Boi13, S.5] definieren den Begriff Informationsraum (information space) als „conceptual tool [...], that can be used to study the codification, abstraction and diffusion of knowledge [...] in a social system“. Somit betrachten sie den Informationsraum vor allem im Zusammenhang mit einem sozialen Lernzyklus und nutzen ihn, um zu verstehen, wie Wissen und Information durch das System fließen.

Benyon [Ben99] zieht dabei die Verbindung zu einer technischen Umsetzung: „The concept of an 'information space' is one that is becoming increasingly important in the current era of networked computers. Information spaces are

manifested in such areas as multiple interacting databases where 'data mining' is a major issue. The same is true in large hypermedia systems such as the world wide web. Even traditional information retrieval systems demonstrate this complexity.“

Heimbigner und McLeod [HM85] betrachten auf einer technischen Ebene zwar nicht den Begriff Informationsraum, aber kommen mit dem Begriff „Federated Database System (FDBS)“ dem Konzept schon sehr nahe. Dabei ist ein FDBS ein System, welches „define[s] the architecture and interconnect[s] databases that minimize central authority yet support partial sharing and coordination among database systems“. Damit haben sie schon früh den Weg zu kompositen Datenbanken gewiesen, die auch für zukünftige Informationsräume ein wichtiges Konzept darstellen werden. Dabei sind verteilte Datenbanken eine Ansammlung an „autonomen Komponenten, die ihre Daten anderen Mitglieder zur Verfügung stellen mit Hilfe einer Veröffentlichung ihres Exportschemas und der Zugriffsmethoden“. Dabei gibt es jedoch kein zusammengeführtes, zentrales Datenbankschema, das die Information von allen Mitgliedern des Verbundes beschreibt.

Als für diese Arbeit am passendsten hat sich die Definition von Hilbert erwiesen[Hil15]: „Ein Informationsraum besteht aus einer semantisch beschriebenen Menge instanzierter oder referenzierter Informationsressourcen sowie zugehöriger semantischer Verknüpfungsinformationen.“ Diese stellt insbesondere heraus, dass die Informationen in den Ressourcen semantisch sind und so formal interpretiert werden können. Weiterhin sind auch die Verknüpfungen zwischen diesen möglicherweise auch heterogenen Informationsressourcen semantisch beschrieben.

### 2.2.2 Data Warehouse

Ein einfaches Mittel, Interoperabilität zwischen verschiedenen Systemen zumindest auf Informationsebene, zu gestalten, besteht in der Einführung eines Datawarehouses (DWH). Als Data Warehouse wird im Allgemeinen eine Datenbank bezeichnet, die für die Durchführung von Analysen optimiert wurde. In einem DWH werden daher Daten aus verschiedenen Quellsystemen integriert und redundant abgelegt. Dadurch bildet das DWH eine effiziente Abfrageschicht, ohne die Performance der Quellsysteme zu beeinträchtigen. Zum Befüllen des DWH, d.h. zum Transport der Daten vom operativen System in die analytische Datenbank, werden spezielle ETL-Werkzeuge (Extract, Transform, Load) verwendet. Ein DWH hat gegenüber der direkten Kopplung von abfragenden Systemen an die einzelnen Datenquellen weiterhin den

Vorteil, dass weniger Schnittstellen zu entwickeln und zu pflegen sind.

Im DWH werden Daten in der Regel in besonderer Weise strukturiert, um einen schnelleren Zugriff bei den häufigsten Abfragen zu erreichen. So können Aggregationen von Kennzahlen und Gruppierungen und Filter in verschiedenen Dimensionen effizient ausgeführt werden. Während es möglich ist, ein DWH basierend auf einem klassischen relationalen Datenbanksystem zu betreiben, werden oft spezielle Lösungen gewählt, um die Abfragen weiter zu beschleunigen. Dazu zählen spaltenorientierte Datenbanken, Key-Value-Datenbanken, dokumentenorientierte Datenbanken sowie In-Memory-Lösungen.

In jüngerer Zeit spielt vor allem das Thema Big Data eine dominierende Rolle, das maßgeblich Einfluss auf die Architektur eines DWH hat. Auf Grund wachsenden Datenaufkommens in Bezug auf Volumen und Geschwindigkeit sind klassische relationale Datenbanken oft überfordert. Ein weiteres Kriterium von Big Data ist häufig die mangelnde Struktur der Daten. Dies legt den Einsatz von NoSQL-Datenbanken<sup>1</sup> nahe, bei denen nicht – wie bei relationalen Datenbanken üblich – ein starres Schema vorzudefinieren ist. Solche Datenbanken optimieren weiterhin oft den Schreibzugriff, was die Echtzeitfähigkeit des Gesamtsystems verbessert. Die Geschwindigkeit neu hinzukommender Daten macht jedoch nicht nur das Ablegen der Daten schwierig, sondern insbesondere auch deren Verarbeitung.

### 2.2.3 Middleware

Middlewares können einen Teil eines Informationsraums stellen. Sie sind nach Tanenbaum und van Steen [TS08] “eine Schicht zwischen Anwendungen und verteilten Plattformen, die in einem gewissen Maß die Verteilung von Daten, die Verarbeitung und die Koordination von verteilten Prozessen vor den Anwendungen verbirgt“. Somit stellt eine Middleware einen Nutzen bei der Entwicklung von Anwendungen dar und betrifft nur indirekt einen Benutzer, der diese Anwendungen einsetzt [VDI11b].

Middlewares stellen eine Informationsschnittstelle zwischen verschiedenen Systemen her, die die Kommunikation von der technischen Umsetzung abstrahiert, so dass sich der Entwickler um die auszutauschenden Informationseinheiten kümmern kann. Anforderungen an Middlewares sind die Abstraktion der Kommunikation, Abbildung des Daten- und Informationsraums, Selbstbeschreibung und Abfragen, Informationssicherheit, Geschäftslogik/Verhalten,

---

<sup>1</sup>Auflistung relevanter Datenbanken unter <http://nosql-database.org/>

Flexibilität (Skalierbarkeit, Erweiterbarkeit, Konfigurierbarkeit), Wartbarkeit, Zuverlässigkeit, Leistung, Zertifizierbarkeit, Interoperabilität [VDI11b].

Ein Vorgehen, das die technologie-unabhängigen Anforderungen zur Informationsübermittlung beschreibt und somit ein neutrales Informationsmodell ohne Spezifika eines Systems ist, kann sowohl die Kommunikation über die Anwendung als auch einen späteren Wechsel einer Middleware gut unterstützen (siehe auch VDI/VDE 2657-2 [VDI15]).

### 2.3 Informationsmodellierung

Sobald verschiedene Informationsträger, was sowohl Menschen als auch Maschinen einschließt, Information austauschen wollen, stellt sich die Frage nach der Übertragung der Information. Dazu gehört nicht nur die Fragestellung, wie technisch auf physikalischer Ebene und Protokoll-Ebene ein gemeinsamer Weg zur Übertragung der Daten mit den enthaltenden Informationen gefunden wird, sondern auch wie die dabei übertragenden Daten zu interpretieren sind.

#### 2.3.1 Semantik vs. Syntax

Üblicherweise wird sich bei dem Informationsaustausch zuerst auf eine gemeinsame Syntax geeinigt (z.B. CAEX, SQL, etc.). Dies ist eine notwendige Bedingung, damit sich die Kommunikationspartner verstehen. Darüber hinaus ist auch eine Einigung über die Semantik der auszutauschenden Daten notwendig.

Dabei wird Semantik häufig als komplementär zu Syntax definiert (z. B. [Däu05]). Während die Syntax die formale Korrektheit von Ausdrücken definiert, gibt Semantik den Ausdrücken ihre Bedeutung. In diesem Sinn geht Semantik in einer sprach-theoretischen Ebene über Syntax hinaus. Syntax definiert dabei die rein formale Struktur einer Sprache. So können auch bei Gebrauch einer korrekten Syntax Sätze in einer Sprache gebildet werden, die keine sinnvolle Semantik enthalten. Ein mögliches Beispiel ist der Satz *Das Wetter öffnet in der Ampel*. Dies ist ein syntaktisch korrekter Satz, aus dem man trotzdem keine sinnvolle Bedeutung ableiten kann.

Semantik kann darüber hinaus auch formal definiert werden (z.B. [Ehr06]). Dies ist jedoch nur sinnvoll zur Nutzung, Erweiterung oder Umdefinition dieser formalen Aspekte. Dieses Unterfangen ist relativ aufwendig, wird jedoch notwendig um tatsächlich die Bedeutung von Ausdrücken maschinell erfassen zu können.

### 2.3.2 Informationsmodell

Ein Modell ist eine „Repräsentation eines Systems von Objekten, Beziehungen und/oder Abläufen. Ein Modell vereinfacht und abstrahiert dabei im Allgemeinen das repräsentierte System“ [Kön12]. Ein Modell bildet natürliche oder künstliche Originale ab, verkürzt diese dabei durch die Fokussierung auf relevante Attribute um somit eine bestimmte Aufgabe pragmatisch erfüllen zu können, in dem es das Original ersetzen kann [Sta73].

Ein Informationsmodell ist ein spezielles Modell, das einen Diskursraum abbildet und eine Einschränkung der Semantik zu Zwecken der Kommunikation darstellt. Dabei wird Informationsmodell nach Lee [Lee99] verstanden als “Repräsentation von Konzepten, Beziehungen, Beschränkungen, Regeln und Operationen, um Semantik für einen gewählten Diskursraum zu spezifizieren”<sup>2</sup>. Ein Informationsmodell kann dabei Vorteile bieten, eine gemeinsam nutzbare, stabile und organisierte Struktur an Informationsanforderungen an den Domänenkontext zur Verfügung zu stellen.

### 2.3.3 Ontologie

Aus einem Informationsmodell kann durch eine formale und explizite Definition eine Ontologie erstellen werden. Somit folgt es der Definition von Studer [SBF98] als “formale, explizite Spezifizierung einer geteilten Konzeptionalisierung”<sup>3</sup>.

Dabei bezieht sich *Konzeptionalisierung* auf ein abstraktes Modell von einem Phänomen, das durch relevante Konzepte dieses Phänomens identifiziert wird.

*Explizit* bezieht sich auf die Definition der genutzten Konzepttypen, ihrer Beziehungen und der jeweiligen Nutzungseinschränkungen.

*Formal* bezieht sich auf die Tatsache, dass eine Ontologie maschinenlesbar sein sollte, was informelle Repräsentationen wie natürliche Sprache ausschließt.

*Geteilt* hat hier die Bedeutung, dass die Ontologie einen Konsens an Wissen reflektiert, also dass es keine private Meinung eines Individuums ist, sondern durch relevante Gruppen akzeptiert wird.

---

<sup>2</sup>Übersetzung des Autors aus dem Englischen “An information model is a representation of concepts, relationships, constraints, rules, and operations to specify data semantics for a chosen domain of discourse”

<sup>3</sup>Übersetzung des Autors aus dem Englischen “An ontology is a formal, explicit specification of a shared conceptualisation”

## 2.4 Modelltransformation

### 2.4.1 Grundlegendes Konzept

In der Realität wird man trotz vielfältiger Bemühungen, relevante Bereiche zu standardisieren, häufig auf die Situation stoßen, dass es unterschiedliche Modelle gibt, die das gleiche Abbild beschreiben, dafür aber unterschiedliche Sichtweisen, Konzepte und Detaillierungsgrade benutzen. Dies ist auch ein sinnvoller Ansatz, da diese Modelle üblicherweise für unterschiedliche Anwendungsfälle gestaltet und genutzt werden und dort genau die ausgewählten Aspekte von Bedeutung sind und die jeweils anderen Aspekte keinen großen Einfluss haben. Nichtsdestotrotz ist es häufig notwendig, dass Modelle, die aus solch unterschiedlichen Meta-Modellen aufgebaut sind, zusammengeführt oder ineinander überführt werden müssen. Wenn dies nicht möglich ist, sollte zumindest eine (Teil-)Abbildung der Konzepte aus dem einen Modell in das andere stattfinden können. Für diesen Zweck eignen sich Modelltransformationen.

Bei Modelltransformation existieren üblicherweise zwei Modelle sowie eine Transformationsdefinition zur Überführung der Modelle ineinander [CH06], wie es auch in Abbildung 2.2 dargestellt ist. Die Definition der Transformation besteht üblicherweise aus einer Reihe von Regeln, die auf den Meta-Modellen des Quell- und Zielmodells definiert sind. Eine Transformationsengine liest das Quellmodell ein, wendet den Regelsatz anhand des Meta-Modells auf das Modell an und erzeugt somit den Zielgraphen. Die aufeinander folgende Anwendung mehrerer Regeln eines Regelsatz kann ein Modell komplett in Einzelschritten in ein neues Modell überführen. In jedem Einzelschritt werden dabei alle Matchings in dem Quellmodell, meist Left-Hand-Side (LHS) genannt, durch ein spezifiziertes Muster ersetzt, wodurch ein neues Zielmodell, meist Right-Hand-Side (RHS) genannt, erzeugt wird.

### 2.4.2 Graph-Transformationen

Eine spezielle Form der Modelltransformation stellt die Klasse der Graphtransformationen dar. Viele Informationsmodelle lassen sich gut als mathematischer Graph abbilden, welcher aus einem Netzwerk aus Knoten und gerichteten Kanten zwischen diesen Knoten besteht [Die12]. Dabei können sowohl Knoten als auch Kanten noch weitere Eigenschaften wie Namen, Klassen, Farben, Gewichte zugeordnet werden, um ein ausdrucksstarkes Modell zu ermöglichen.

Graph-Transformationsansätze nutzen die spezielle Struktur von Graphen und erlauben damit die Gestaltung von mächtigen Transformationsregeln.

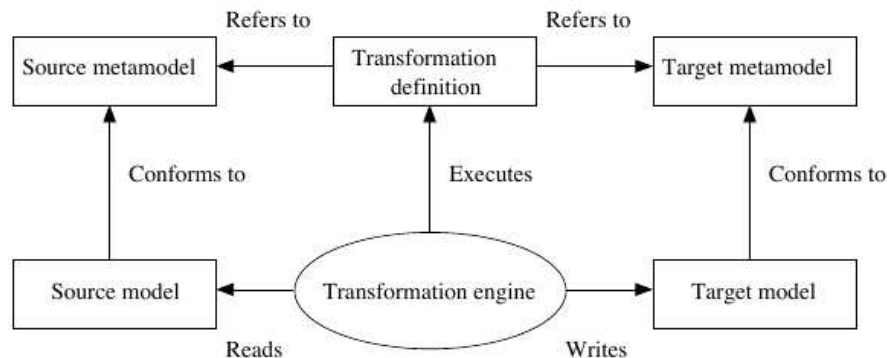


Abbildung 2.2: Modelltransformation [CH06]

Im Gegensatz zu Transformationen auf Text-Ebene ist es hiermit leichter, Abhängigkeiten und Beziehungen zwischen Konzepten in die Transformationsregeln einzubringen. Zudem können die Regeln auf eine deklarative Weise spezifiziert werden, was eine hohe Wiederverwendbarkeit und Wartbarkeit verspricht.

#### 2.4.2.1 Graph-Transformationsansätze

Obwohl die Grundstruktur eines Graphen sehr simpel ist, gibt es eine ganze Reihe von Graphtransformationsansätzen [Tae+05]. Das liegt unter anderem daran, dass ein Satz von Regeln ohne weitere externe Bedingungen häufig nicht deterministisch sind. So können sowohl mehrere Passungen einer Regel als auch mehrere Regeln gleichzeitig anwendbar sein. Hier muss eine Entscheidung über Reihenfolge und Parallelität getroffen werden.

Bei der Anwendung der Regeln können zudem ungewollte Seiteneffekte auftreten, so dass ein Knoten gelöscht werden soll und gleichzeitig aber noch eine weitere Kante auf diesen Knoten verweist. Die Priorisierung der Löschung von Knoten und Kanten führt zu dem Ansatz des Single-Push-Out (SPO). Hier wird im Gegensatz zum Ansatz des Double-Push-Out (DPO) nicht geprüft, ob durch die Anwendung der Regel ein Identifikations- oder Kontaktproblem entsteht. Diese würden bei DPO zu einer Nicht-Anwendbarkeit der Regel führen.

Darüberhinaus gibt es noch erweiterte Ansätze, die zusätzliche Konzepte einführen. Dazu wird im nächsten Kapitel ein spezieller Ansatz mit dem Namen Triple-Graph-Grammars detailliert vorgestellt.

### 2.4.2.2 Triple Graph Grammars

Triple Graph Grammars (TGG) ist ein spezieller Graph-Transformationsansatz, in dem die Transformation in beide Richtungen durch eine einzige Regel ausgedrückt werden kann [Sch94]. Mit Hilfe dieser Regel ist eine Prüfungen möglich, ob zwei Modelle noch konsistent sind, und die Anwendung synchroner Operationen auf beide Modelle möglich.

Dazu wird für Transformation neben den üblichen Domänen der beiden Modelle noch eine Kontext-Ebene eingeführt, die die Verbindung der Elemente der beiden Modellwelten enthält (siehe Abbildung 2.3). Nun wird deklarativ spezifiziert, was eine konsistente Änderung aller drei Bereiche sein soll (in der Abbildung durch das Keyword «new» gekennzeichnet). Abbildung B.3 im Anhang zeigt die Ableitung von operationalen Regeln aus dieser deklarativen Regel. Mit diesen können jeweils eine Transformation von links nach rechts, von rechts nach links, eine Korrespondenzprüfung oder einer Modellsynchronisierung durchgeführt werden.

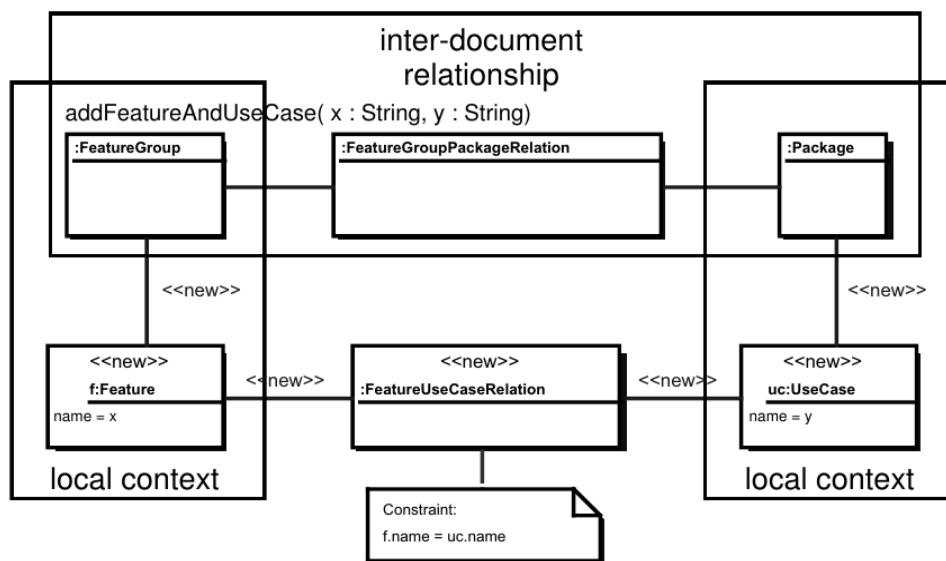


Abbildung 2.3: Deklarative TGG Regel [KS06]

### 2.4.2.3 Anwendungsfälle für Triple Graph Grammars

Kindler [KW07] beschreibt drei Anwendungsfälle für Transformationen mit TGG, die sich jedoch auch allgemein auf andere Transformationsansätze



anwenden lassen.

**Modell-Transformation** Der offensichtlichste Anwendungsfall ist die Transformation eines Modells in ein anderes. Bei dieser existiert genau eins der beiden Modelle, das sogenannte Quell-Modell. Mit Hilfe von Transformationsregeln wird daraus ein korrespondierendes Modell erzeugt.

**Modell-Integration** Im zweiten Fall gibt es bereits beide Modelle und der Nutzer möchte die Korrespondenz zwischen beiden erfahren. Dazu ist die Erzeugung von Beziehungen der einzelnen Elemente zu den Elementen aus der anderen Domäne zu erstellen.

**Modell-Synchronisation** Auch beim letzten Anwendungsfall existieren beide Modelle sowie die Korrespondenzen zwischen beiden Modellen. Nun ist hier die Aufgabe, beide Modelle parallel zu ändern, so dass diese kongruent bleiben. Damit sind inkrementelle Transformationen möglich, die Änderungen des abgebildeten Objekts in beiden Modellwelten nachführen können. Dies ist ein häufiger Anwendungsfall für das Anwendungsgebiet dieser Arbeit.



## 3 Anforderungsanalyse

### Aufbau und Einordnung des Kapitels

In diesem Kapitel werden anhand einer Literaturrecherche wichtige Anforderungen an zukünftige Informationsräume extrahiert und klassifiziert. Dazu wird zuerst eine Einordnung des Fokus der Literaturrecherche vorgenommen und das Vorgehen zur Ableitung und Strukturierung von Anforderungen beschrieben. Danach erfolgt eine Beschreibung der abgeleiteten Anforderungen in drei Gruppen.

### 3.1 Vorgehen zur Anforderungsermittlung

Um geeignete Anforderungen an zukünftige Informationsräume zu extrahieren, wird eine Literaturrecherche durchgeführt. Als Startpunkt eignet sich eine Ausführung von Schleipen, Münnemann und Sauer, die einen guten Überblick über notwendige Anforderungen an zukünftige Systeme gibt [SMS11]:

- Hohe Marktdynamik und damit hohe Änderungsgeschwindigkeit, z. B. bei Kundenaufträgen,
- Zunehmende Produktdiversifikation und hohe Variantenvielfalt,
- Kurze Lieferzeiten und schnelle Reaktionen auf Kundenwünsche,
- Steigende Anforderungen an Produkt- und Prozessqualität und deren Dokumentation,
- Minimierung des Ressourcenverbrauchs, z. B. Energie- und Rohstoffeffizienz,
- Optimale Anlagenauslastung bzw. Skalierbarkeit von Kapazitäten,
- Zeitliche Engpässe von qualifiziertem Personal und dessen effizienter Versorgung mit Informationen.

Diese generellen Anforderungen müssen jedoch auf technische Aspekte heruntergebrochen werden. Um einen breiten Fokus zu behalten und die Anwendbarkeit auf möglichst viele Szenarien gewährleisten zu können, wird keine Fokussierung auf bestimmte Anwendungsfälle vorgenommen, sondern allgemeine Aspekte heutiger und zukünftiger Entwicklungen betrachtet.

Der Ansatz einer größeren Vernetzung in Produktionsnetzwerken hat unter dem Stichwort Industrie 4.0 in Deutschland und als Advanced Manufacturing, Cyber-Physical-Systems und ähnliches weltweit große Beachtung gefunden. In den letzten Jahren sind eine Vielzahl an Positionspapieren und Artikeln mit Ausblick in die Zukunft erschienen (z.B. [Pri14; SHS14; Spa+13; Fa13; Pla15; VDI13]). Diese werden anhand ihrer Aussagen zu Anforderungen an Informationsräume untersucht und hier zusammengefasst.

Daneben werden ausgewählte nationale und internationale akademische Publikationen betrachtet, die Aussagen über Informationsräume machen (z.B. [Sch07; Mah+11] sowie Richtlinien und Normen (z.B. [VDI11a; NAM14]).

Dazu lassen sich die in der Literatur aufgenommenen Anforderungen in die drei bewährten Bereiche *Beschreibungsmittel*, *Methoden*, und *Werkzeuge* der Betriebsmittel für das Systemengineering einteilen [Sch13; CJS98]. Dies soll eine einfache Strukturierung und Betrachtung ähnlich gearteter Anforderungen ermöglichen. Für jede Anforderung wird eine Definition der wesentlichen Eigenschaften aufgestellt, bevor diese aus Quellen abgeleitet und in ihrer Wichtigkeit nachgewiesen wird.

Geeignete Anforderungen zeichnen sich dadurch aus, dass diese eine differenzierende Eigenschaft zwischen Informationssystemen aufweisen und messbar ist. Jedoch können nicht alle Anforderungen trennscharf in einen dieser Bereiche aufgeteilt werden. Beispielsweise werden für eine Revisionierung sowohl Beschreibungsmittel zur Ablage von Versionsinformationen, als auch mathematischen Methoden für eine effiziente Komprimierung sowie letztendlich Werkzeuge, die diese Funktionalität bereitstellen, benötigt. In diesen Fällen wird die entsprechende Anforderung in den Bereich mit den größten Auswirkungen eingeteilt.

## 3.2 Anforderungen an industrielle Informationsräume

### 3.2.1 Beschreibungsmittel

Der Bereich Informationsmodellierung beschreibt die Anforderungen, die zukünftige Informationssysteme für die Industrie erfüllen müssen, um den auszutauschenden Informationshaushalt detailliert spezifizieren und verwalten

zu können. Desweiteren fallen in diese Kategorien auch die notwendigen Voraussetzungen, um organisationsübergreifend flexibel zusammenarbeiten zu können.

### 3.2.1.1 Mächtige Informationsmodellierung (B1)

*Die Informationsmodelle, die für die Abbildung der digitalen Anlage notwendig sind, weisen eine hohe Komplexität auf, da die abzubildende Wirklichkeit der realen Anlage hochkomplex ist. Daher muss die Technologie eine ausreichende Mächtigkeit in der Informationsmodellierung besitzen, so dass die gewünschten Konzepte möglichst einfach abgebildet werden können. Daher sollen die Informationsmodelle eine hohe Flexibilität ohne größere technische Einschränkungen aufweisen.*

Mahnke et. al. [Mah+11] zeigen, dass sich verschiedene Middlewares und Informationssysteme in ihren Konzepten für die Informationsmodellierung in der Mächtigkeit unterscheiden und damit unterschiedlich gut komplexe Informationen abbilden können. Es ist zwar auch mit einfachen Modellierungskonzepten, wie Attribute und einfache Datentypen, komplexe Originale abzubilden; jedoch ist die Modellierung mit der Verwendung von mächtigeren Konstrukten wie Objektorientierung, Vererbung und Referenzierung weniger aufwändig und wartbarer durchzuführen.

Laut Schlund, Hämmerle und Strölin [SHS14, S. 12] sind massive Investitionen in die Daten-Qualität notwendig, um Industrie 4.0 erfolgreich umzusetzen. „Industrie 4.0-Anwendungen benötigen eine aktuelle, für die jeweilige Nutzung genaue und verlässliche Datengrundlage. [...] Herausforderungen im Bereich der Datenqualität betreffen derzeit beispielsweise die Genauigkeit, Vollständigkeit und Aktualität [...]“

### 3.2.1.2 Semantik (B2)

*Über die durch die Kommunikationsschnittstellen der Middleware bereitgestellten Dienste werden Daten zwischen den Teilnehmern ausgetauscht. Dabei ist es Aufgabe der Middleware zu definieren, welche Daten wie ausgetauscht werden. Neben der Syntax, die (häufig transparent für die Nutzer der Middleware) das Format der Daten festlegt, muss eine Aussage über die Semantik der Daten getroffen werden. Je nach Anwendung ist hier von Bedeutung, ob z. B. die Middleware ein generisches oder ein konkretes Modell festlegt oder das Modell erweiterbar ist.[VDI11b]*

Middlewareansätze, die Abstraktionsmechanismen anbieten, mit denen

Kontext und Semantik durch die Betreiberorganisation modelliert und abgebildet werden können, sind aufgrund der hohen Anforderungen an Flexibilität und Anpassbarkeit bei langen Laufzeiten für die Aufgaben der Automatisierungstechnik besser geeignet als Middleware-Lösungen mit durch den Hersteller festgelegter Abstraktion [VDI11b]. Für die Akzeptanz ist es wichtig, dass die Semantik hersteller-neutral ist [NAM14].

#### 3.2.1.3 Erweiterbarkeit (B3)

*Die Modelle in Informationssystemen haben ebenso wie industrielle Prozesse und Anlagen einen Lebenszyklus, da sie diesen abbilden. Daher ist es notwendig, dass sich die Informationsmodelle ohne großen Aufwand flexibel an neue Gegebenheiten anpassen und erweitern lassen. Dies soll zudem auch zur Laufzeit geschehen können. Diese Anforderung wird sich insbesondere in volatilen Märkten verstärkt ausdrücken.*

Die Dissertation von Stephan [Ste12, S. 3] sieht „Wandelbarkeit als zentrales Merkmal in der Gestaltung von Fabriken und Produktionssystemen“ – auch über den Aspekt der reinen Instandhaltung hinaus. Ebenso fordert der VDI/VDE [VDI13] in seinem Papier zu Cyber-Physical Systems, dass „zur Entwurfszeit nicht vorhersehbare Änderungen im Betrieb mit CPS (z. B. nachgeladene Applikationen, veränderte Systemtopologien) [...] problemlos integriert werden können“ müssen, was vor allem den Aspekt der Erweiterbarkeit zur Laufzeit betrifft. Auch die VDI/VDE 2657-1 [VDI11b] und die NE 150 [NAM14] stellen die Bedeutung der dynamischen Erweiterbarkeit des verwendeten Informationsmodells heraus.

Fricke und Schulz [FS05] stellen bei der Betrachtung von *Design for Changeability*, welches sie als Kernanforderung für komplexe Systeme sehen, folgende vier Aspekte heraus: Robustheit, Flexibilität, Agilität und Adaptierbarkeit. Damit sich das Gesamtsystem ändern und an neue Gegebenheiten anpassen kann, können die Punkte Flexibilität, Agilität und Adaptierbarkeit für hochgradig digitale Systeme durch eine Erweiterbarkeit des Informationsmodells erreicht werden. Auch [Spa+13, S. 74] sieht auf Grund von volatilen Märkten und Mitarbeiterorientierung Flexibilität als Schlüsselfaktor, die wiederum in den Informationsmodellen abgebildet werden muss.

#### 3.2.1.4 Selbstbeschreibungsfähigkeit (B4)

*Selbstbeschreibungsfähigkeit sowie deren Abfragbarkeit sind ein wichtiger Bestandteil, um die durch die Abstraktion gewonnene Unabhängigkeit der Be-*

*schreibung und Daten nutzen zu können. Angebotene Dienste basieren nicht nur auf Quellen und Informationen des Systems selbst, sondern liefern auch Zugriff auf abgebildete Informationen weiterer Komponenten, was bedeutet, dass diese Dienste sich dynamisch zur Laufzeit ändern können. Um dennoch auf diese Informationen zugreifen zu können, ist eine Abfrage dieser Dynamik eine Voraussetzung für einen konsistenten Informationsraum.*

Für diese Beschreibung wurde in großen Teilen auf die VDI/VDE 2657-1 [VDI11b] zurückgegriffen und deren Anwendung von Middlewares auf allgemeine Informationsräume erweitert. Die Richtlinie sieht eine Möglichkeit zur Realisierung von Selbstbeschreibungsfähigkeit in der Bereitstellung von Meta-Modellen. Diese Fähigkeit der Selbstbeschreibung einer aktuellen Struktur kann jedoch auf vielen Ebenen realisiert werden.

So ist Selbstbeschreibungsfähigkeit eine notwendige Maßnahme, um die Anforderung von [VDI13] „zur Entwurfszeit nicht vorhersehbare Änderungen im Betrieb mit CPS (z. B. nachgeladene Applikationen, veränderte Systemtopologien) problemlos [integrieren zu] können.“

Ähnlich sieht es [AG212], die „zusätzlich zum Daten- und Informationsaustausch eine standardisierte Erkennung der Geräte und Beschreibung der Funktionalität“ für erforderlich hält.

#### **3.2.1.5 Organisationsübergreifende Verknüpfungen (B5)**

*Die Vernetzung von Informationen aus unterschiedlichen Datenbehältern ist eine grundlegende Anforderung an Informationsräume. In Zukunft wird sich diese Anforderung auf die Vernetzung über Unternehmensgrenzen hinweg ausweiten. Dazu ist eine globale Identifizierung einzelner Informationsentitäten notwendig.*

So sieht Schack in seinen Herausforderungen und Trends für die digitale Anlage auch die „vernetzte unternehmensinterne und -übergreifende Zusammenarbeit“ [Sch07, S. 39]. Die Betrachtung konzentriert sich dort auf eine organisatorische Ebene. Diese muss jedoch durch geeignete Mittel auch technisch unterstützt werden.

Der Statusreport Industrie 4.0 vom VDI/VDE von 2014 [VDI14] sieht Interoperabilität als entscheidendes Erfolgskriterium in einem System von Systemen. Dabei ist hier explizit von unternehmensübergreifenden und domänenübergreifenden Verknüpfungen die Rede.

### 3.2.2 Methoden

Der Bereich Methoden beschreibt die technologischen Anforderungen, die zukünftige Informationssysteme für die Industrie erfüllen müssen, um sich in vorhandene Systeme integrieren und mit anderen Systemen sicher und performant austauschen zu können.

#### 3.2.2.1 Abstraktion der Kommunikation (M1)

*Ein zukünftiger Informationsraum übernimmt gemeinsame Aspekte und Dienste der Kommunikation, um Komponenten von diesen zu entlasten und eine einheitliche Kommunikationsschnittstelle anzubieten. Wesentliche Aufgaben sind dabei das Management der Kommunikation, wie Verbindungsaufbau oder Verschlüsselung, und die Bereitstellung von spezifischen Diensten, wie Datenaustausch oder Methodenaufruf, unabhängig von der technologisch bereitstehenden Realisierung. Die konkrete Umsetzung auf die Kommunikationsmechanismen des eingesetzten Mediums ist dabei für die Anwendung transparent.*

Diese Definition der Kommunikationsabstraktion leitet sich direkt aus den Anforderungen von Middlewares her [VDI11b]. Gestützt wird dieser Punkt dabei durch [AG212], laut der „die Vielzahl der Übertragungstechniken (Kabel, WLAN, GSM, GPRS) und Protokolle (basierend auf TCP, HTTP, seriell, Feldbus, etc.) [...] die transportunabhängige, durchgängige „Interoperabilität“ notwendig [macht].“

#### 3.2.2.2 Statische Daten (M2)

*Die Abbildung und der Transfer von statischen Daten nimmt vor allem in der Planung einen großen Stellenwert ein. Daher müssen zukünftige Automatisierungssysteme Mechanismen bereitstellen, um Daten für eine einfache Datenhaltung und Datenaustausch zu serialisieren und persistieren zu können.*

So erfordern neue wettbewerbsfähige Lösungen laut Schack [Sch07, S. 39] eine „durchgehende und integrierte Verwendung von Planungsdaten“. Dies wird durch die VDI 4499 [VDI11a, S. 25] unterstützt, die geeignete Systemarchitekturen und Datenmanagementsysteme zur Vernetzung von Modellen fordert.



### 3.2.2.3 Dynamische Daten (M3)

*In der Betriebsphase werden die statischen Daten durch dynamische Daten ergänzt. Auch diese müssen zukünftige Informationsräume performant verwalten können. Durch eine Verknüpfung mit den statischen Planungsdaten sind mächtige Auswertungsalgorithmen möglich.*

Der Betrieb einer Anlage erfordert die Auswertung aktueller Daten, die von Sensoren oder anderen Systemen bereit gestellt werden. Diese entstehen in Echtzeit und stellen ein Abbild der realen, physikalischen Welt dar [AG212]. „Durch Vernetzung mit internetbasierten Diensten“ [AG212] können diese für autonome Regelprozesse genutzt werden. Allgemein, müssen sich dynamische Daten in den gesamten Informationsraum möglichst nahtlos integrieren, um eine konsistente Abfragemöglichkeit in Verbindung mit den statischen Daten aus der Planung zu ermöglichen.

### 3.2.2.4 Informationssicherheit (M4)

*Die Sicherheit von Informationsräumen besteht einerseits darin, den gesamten Informationsraum vor äußeren Angriffen zu schützen. Andererseits müssen auch innerhalb des Informationsraumes Mechanismen verfügbar sein, die eine feingranulare Autorisierung für Informationsentitäten ermöglichen.*

Eine immer wieder auftauchende Forderung für zukünftige Systeme ist, dass „die Sicherheit CPS-basierter Automatisierungslösungen [...] auf dem hohen Standard der heutigen Automation gewährleistet bleiben“ muss [VDI13]. Dies wird ebenso von [Pla15] („Ohne eine umfassende Gewährleistung des Datenschutzes sowie der Informationssicherheit wird eine verteilte Datenhaltung in der Industrie 4.0 kaum zu realisieren sein“), [Fa13] sowie [DD15] und [NAM14] bestätigt.

Hierbei wird zwischen der funktionalen Sicherheit (Safety) und Informationssicherheit (Security) unterschieden. Die funktionale Sicherheit gewährleistet den Schutz von Mensch, Umwelt und Anlage beim Betrieb von technischen Systemen [LG13]. Informationssicherheit stellt die Verfügbarkeit, Integrität und Vertraulichkeit der Informationen in den Anlagen und Systemen sicher [Pla15]. Bei Security geht es darum, Gefahren abzuwehren, die auf die Anlage bzw. deren Funktionen einwirken. Insbesondere sind explizite und nicht-intendierte Angriffe eingeschlossen. Dies bedeutet in vernetzten Produktionsnetzwerken vor allem auch ein Schutz vor dem Abfließen von Wissen, das in Daten vorgehalten wird. Da dennoch ein Austausch von Informationen stattfinden soll, ist dies nur durch feingranulare Autorisierungs- und

Zugriffsmethoden zu erreichen.

#### 3.2.2.5 Dienstorientierung (M5)

*Einzelne Komponenten in zukünftigen Automatisierungssystemen werden zunehmend Dienstleister sein. Daher muss der Informationsraum auf diese Dienst-Orientierung vorbereitet sein und diese unterstützen.*

Dienste werden in Industrie 4.0 Szenarien sowohl auf der Feldebene als auch den darüber liegenden Ebenen eine größere Rolle spielen [Die+14]. Dabei ist auch eine semantische Beschreibung der Dienste, deren Parameter und Rückgabewerte ein entscheidender Schritt, um eine größtmögliche Interoperabilität herstellen zu können.

Im Zuge der Modularisierung von verfahrenstechnischen Anlagen wandert ein Großteil der Steuerungs- und Automatisierungsaufgaben direkt in die eigenständigen Module in einer Anlage [Hol+14]. Für eine koordinierende Steuerung der Module soll keine komplette Integration aller Signale in ein übergeordnetes Leitsystem stattfinden, weil damit Vorteile der Flexibilität und Agilität wieder aufgegeben werden. Vielmehr bieten die Module Dienste an und kümmern sich selbst um die Erfüllung der darin definierten Aufgabe [Obs+15a; Obs+15b]. Diese Dienste müssen über geeignete technische Schnittstellen angesteuert werden können.

#### 3.2.3 Werkzeuge

Der Bereich Werkzeuge beschreibt die Anforderungen, die zukünftige Informationssysteme für die Industrie erfüllen müssen, um sich in die Prozesse von flexiblen Wertschöpfungsnetzwerken einfügen zu können.

##### 3.2.3.1 Modelltransformationen (W1)

*Eine komplette Vereinheitlichung von Datenmodellen ist auf Grund unterschiedlicher Philosophien in Unternehmen und Universitäten ein Ziel, das nur in eng begrenzten Bereichen funktioniert. Zur Herstellung einer Interoperabilität ist somit eine Transformation zwischen ähnlichen Datenmodellen notwendig.*

Damit unterschiedliche Organisationen interagieren können, ist die Abbildung der jeweils genutzten Konzepte auf die der anderen Organisation notwendig. Dies kann einerseits durch eine eindeutige Standardisierung der Komponenten gelöst werden, zum anderen aber auch durch eine Abbildung der einzelnen Modellteile, die dann durch eine Modelltransformation ausgeführt

werden kann. In der Kollaboration von verschiedenen Organisationen ergibt sich fast immer die Situation, dass jede Organisation bestimmte Teilaspekte anders modelliert oder gewisse Funktionalitäten exklusiv anbietet. Dadurch ergeben sich für den reinen Standardisierungsansatz große Hindernisse, auch wenn nur zwischen zwei Beteiligten vereinheitlicht werden soll und nicht gleich ein allgemein akzeptierter internationaler Standard ausgearbeitet werden muss. Denn bis die neuen und innovativen Funktionalitäten standardisiert sind und somit eine Zusammenarbeit für diese möglich ist, vergeht viel wertvolle Zeit. Desweiteren bedeutet eine Standardisierung, dass sich mindestens ein Partner sein eigenes Modell anpassen und auf das neue umstellen muss, was wiederum eine Umstellung aller vorhandenen Werkzeuge bedeutet. Hier bietet sich als erster Schritt eine Modelltransformation auf das Modell des Partners oder ein gemeinsames Zwischenmodell an. Aber auch innerhalb von Unternehmen ist eine schnelle Zusammen- und Rückführung von Daten ein wesentliches Mittel zur Produktivitätssteigerung [Spa+13], das unter anderem durch Modelltransformationen zu erreichen ist. Modelltransformation wird auch explizit durch [Pla15] als Aufgabe des Information-Layers gefordert, um die Industrie 4.0-Komponenten aufeinander abbilden zu können.

#### 3.2.3.2 Revisionierung (W2)

*Gerade in dezentralen Wertschöpfungsnetzwerken ist es erforderlich, eine Versionsverwaltung von Informationen aufzusetzen, die es erlaubt, bewusst auf unterschiedlichen Versionsständen arbeiten zu können. Hierbei ist eine enge Einbindung des Zugriffs auf alte Revisionen notwendig, der idealerweise keine weitere Schnittstelle benötigt. Weiterhin ist auf einen schnellen und sicheren Zugriff zu achten.*

Laut der Analyse von Schmidt et. al. ist Revisionierung eine der Hauptanforderungen an industrielle Informationssysteme [Sch+14a]. Diese ist laut der dort durchgeführten Studie sogar wichtiger als ein konsistenter Datenaustausch und automatische Konsistenzprüfungen. Auch die NAMUR [NAM14] sieht Änderungsmanagement als zentrale Anforderung.

Ähnlich wird dies in der Analyse von Braun [Bra+12, S. 340-345] gesehen. Um die spezifischen Anforderungen an Variabilität und Integrität in der Evolution von Automatisierungssystemen gerecht zu werden, gibt es die Forderung nach: 1) „Integrated variant- and version-management of product lines (Application Modules) and a product line evolution engineering process with respect to adaptation of product instances in the field“ und 2) „multi-dimensional (within a single discipline and interdisciplinary) module and

interface concept and a corresponding compatibility validation process.“

#### **3.2.3.3 Werkzeugunterstützung für die Entwicklung (W3)**

*Gerade für die Entwicklung von Applikationen ist eine gute Werkzeugunterstützung notwendig, die dem Nutzer einen Teil der Arbeit abnehmen. So sind Werkzeuge notwendig, mit denen Informationsstände visualisiert, untersucht und miteinander verglichen werden können. Auch eine Unterstützung durch umfangreiche Bibliotheken von einer Reihe von Programmiersprachen und Plattformen ist eine enorm wichtige Hilfestellung für den Nutzer.*

Ein Informationsraum wird nur dann erfolgreich und weit verbreitet genutzt werden, wenn die entsprechenden Werkzeuge und Bibliotheken vorhanden sind, um mit diesem auf einfache Art und Weise zu interagieren. Zur Applikationsentwicklung in diesem Informationsraum sind über die reinen Bibliotheken in einer Programmiersprache der Wahl auch weitergehende Werkzeuge zum Debugging notwendig, um eine schnelle und qualitativ hochwertige Anwendungsentwicklung zu ermöglichen.

#### **3.2.3.4 Anbindung vorhandener Daten (W4)**

*Ein wichtiges Merkmal ist die Wiederverwendung von vorhandenen Daten. Dazu bildet dieses Merkmal die Fähigkeit von Informationsräumen ab, vorhandene Daten aus anderen Systemen durch entsprechende Importschnittstellen, Adapter oder andere Mechanismen einzubinden.*

So müssen laut Schack [Sch07, S. 39] eine „durchgehende und integrierte Verwendung von Planungsdaten“ erfolgen, die durch offene Systemlösungen mit entsprechenden Schnittstellen umgesetzt werden müssen. Die vorhandenen Daten sollen universell und anwendungsunabhängig einsetzbar sein und damit eine Wiederverwendbarkeit ermöglichen [NAM14].

#### **3.2.3.5 Unterstützung zur Nachverfolgbarkeit/Inspizierbarkeit zur Laufzeit (W5)**

*In verteilten Informationsräumen wird es wichtig sein, den aktuellen Informationsstand sowie mögliche Änderungen dem Nutzer nachvollziehbar zu präsentieren. Gerade durch die enorme Komplexität benötigt der Nutzer eine gute Werkzeugunterstützung, um Entscheidungen des Systems nachvollziehen und bewerten zu können. Dafür sind geeignete Visualisierungen und Eingriffsmethoden erforderlich.*

Auch nach der Erstellung von Anwendungen in einem verteilten Informationsraum ist eine Nachvollziehbarkeit der Informationen und Entscheidungen in einer Applikation essentiell. Der Nutzer der Anwendung sollte die Nachvollziehbarkeit der Entscheidungen der Applikationen durch diese selbst in geeigneter Weise visualisiert bekommen. Durch die Verteilung des Informationsraums können nämlich für den Nutzer unerwartete Informationsflüsse auftreten, die durch Änderungen in einem entfernten Datensatz ausgelöst werden. Wenn sich jedoch darüber hinaus auch das Informationsmodell eines Datensatzes ändert, muss dem Entwickler eine Möglichkeit gegeben sein, diese nicht durch die Applikation abgebildeten Änderungen nachvollziehen und entsprechend darauf reagieren zu können.



## 4 Analyse bestehender Ansätze

### Aufbau und Einordnung des Kapitels

Das folgende Kapitel analysiert existierende Ansätze, die für den Aufbau und den Betrieb von Informationsräumen genutzt werden können. Dazu werden die Ansätze beschrieben und hinsichtlich der Erfüllung der Anforderungen aus Kapitel 3 bewertet. Dabei werden Ansätze aus der Automatisierungstechnik (OPC UA, AutomationML) der Verfahrenstechnik (ISO 15926) und der Informatik (WBEM/CIM, Eclipse EMF, Semantic Web) diskutiert.

### 4.1 Semantic Web

Das Semantic Web stellt ein Framework bereit, das es erlaubt Daten über Grenzen von Applikationen, Unternehmen und Communities hinweg zu teilen und wiederzuverwenden [W3C11]. Dabei wird es hauptsächlich vom W3C mit einer breiten Unterstützung durch akademische und industrielle Partner vorangetrieben. Als Basistechnologien wird auf das Resource Description Framework (RDF), SPARQL zur Abfrage von Informationen, Internettechnologien wie HTTP (Hypertext Transfer Protocol) und Ontologien aufbauend auf RDFS und OWL zurückgegriffen. Diese Technologien lassen sich auch unter dem Stichwort *Linked Data* subsummieren [BHB09; Biz09]. Eine detaillierte Beschreibung der Technologien und wichtiger Werkzeuge ist im Anhang im Kapitel C auf Seite 125 zu finden.

Linked Open Data ist dabei der Teil der Daten, die öffentlich und frei verfügbar zugänglich sind. Linked Open Data weist ein rasante Wachstumskurve mit einer Verdreifachung der Datenbasis in den letzten Jahren [iX14] auf. Dabei gibt es eine große Vernetzung zwischen den einzelnen Datensätzen, wobei einige Datensätze große Hubs darstellen, wie z.B. Dbpedia, Freebase und Geo Names.

#### 4.1.1 Ziele

Die Ziele, die das Semantic Web verfolgt, orientieren sich an den Zielen des World Wide Web, das auf eine starke Offenheit und Einfachheit setzt.

**Einfache Erweiterbarkeit** Ein großer Vorteil von Linked Data liegt in seiner einfachen Erweiterbarkeit. Diese kann durch die *Open World Assumption* [Ber09] erreicht werden, die alle nicht explizit formulierten Aussagen als nicht prüfbar annimmt. Das steht im Widerspruch zu *Closed-World*-Systemen, die solche Aussagen als unwahr bewerten würden.

**Model as you use** Ein wesentlicher Ansatz des Semantic Web ist, die Initialinvestition möglichst gering zu halten. So sollten am Anfang nur die wirklich notwendigen Konzepte modelliert werden. Die weiteren Aufwände für eine Modellausweitung oder Neugestaltung sind somit erst zu dem Zeitpunkt zu erbringen, an dem sie auch einen direkten Nutzen bringen. Somit ist ein schneller Einstieg gewährleistet.

**Semantisches Informationsmodell** Die Informationen, die auf Webseiten zu finden sind, sollen durch Maschinen interpretierbar sein. Dazu ist eine semantische Beschreibung der verwendeten Konzepte notwendig, die sich idealerweise in dem gleichen Informationsmodell wiederfinden.

**Dezentrale Architektur** Wie im bisherigen Web soll die Architektur verteilt bleiben, so dass eine Aufteilung der Information auf verschiedene Server vollzogen wird. Auch ansonsten soll es keinen Master-Knoten geben, sondern die verschiedenen Server sollen prinzipiell gleichrangig sein. So können sich neue Nutzer einfach mit der Integration eines Servers in das Netzwerk integrieren.

**Leichtgewichtiger Ansatz** Die Linked Data Philosophie ist derjenigen von Linux sehr ähnlich und hält viele spezialisierte Werkzeuge vor, deren clevere Kombination mächtige Anwendungen zu gestalten erlaubt. Auf der anderen Seite kann durch die Beschränkung auf einige wenige Werkzeuge, der Einarbeitungs- und Wartungsaufwand gering gehalten werden. Desweiteren ist durch diese Leichtgewichtigkeit der einzelnen Werkzeuge schnell eine Weiterentwicklung eines einzelnen Werkzeugs möglich.

**Domänenübergreifende Vernetzung** Eine Hauptstärke liegt in der Wiederverwendung von bereits modellierten Informationen. Dazu werden Verknüpfungen auch über Domänengrenzen als First-Class-Konzepte aufgefasst. Das ermöglicht es, Informationsmodelle aufzuteilen und mit den entsprechenden Konzepten weiterzuentwickeln, um so einen möglichst großen und



umfassendes Gesamtwissensnetzwerk aufzubauen. Dabei erzeugen zusätzliche Verknüpfungen mehr Kontext und steigern somit den Wert der Information.

### 4.1.2 Prinzipien

Diese Ziele sollen durch einige wenige Prinzipien erreicht werden, die Tim Berners-Lee [Ber06] 2006 als Grundprinzipien von “Linked Data” aufgestellt hat:

1. Use URIs to identify things.
2. Use HTTP URIs so that these things can be referred to and looked up (“dereferenced”) by people and user agents.
3. Provide useful information about the thing when its URI is dereferenced, using standard formats such as RDF/XML.
4. Include links to other, related URIs in the exposed data to improve discovery of other related information on the Web.

Das Semantic Web ist eine mögliche Implementierung dieser Linked Data Prinzipien. Prinzipiell können diese Prinzipien und vor allem deren Verallgemeinerung durch [zaz11a] aber auch mit anderen Technologien erreicht werden.

### 4.1.3 Best Practises

Einen gut nutzbaren Informationsraum lässt sich nicht nur durch eine Festlegung auf die vier Basisprinzipien und eine Reihe von Basis-Technologien erreichen. Man benötigt viel mehr eine stärkere Unterstützung der Entwickler beim Aufsetzen der Systeme, um den Nutzern eine leichte Orientierung und Wiedererkennungswerte zu geben. Dazu haben sich einige zusätzliche Best Practises herauskristallisiert, die versuchen die offene Natur des Semantic Web etwas stärker einzugrenzen, wobei der Informationsraum auch ohne Einhaltung dieser Richtlinien als solcher nutzbar und wertvoll bleibt.

Einen wichtigen Teil nehmen davon Grundregeln für die Auswahl und das Erstellen geeigneter URIs ein<sup>1</sup>. Insbesondere auf die Aufspaltung von Informationsdarstellung als HTML und als RDF-Repräsentation wird näher eingegangen und wie richtig auf diese verwiesen wird, so dass nach einer Content-Negotiation immer noch die richtigen Inhalte ausgeliefert werden.

---

<sup>1</sup><http://www.w3.org/TR/cooluris/>

Das *LATC (Linked Open Data Around-The-Clock)*<sup>2</sup> Projekt ist eine Unterstützungsplattform für das FP7 Programm, das bei dem Veröffentlichen und Konsumieren von Linked Data hilft. Dazu hat es eine Reihe von Richtlinien herausgegeben. Eine der hilfreichsten ist dabei die *LATC Dataset Publishing guideline*<sup>3</sup>.

### 4.2 ISO 15926

Die ISO 15926 [ISO07] ist ein internationaler Standard für die Integration und den Austausch von Daten. Initial als Nachfolger von STEP [Fow95] gedacht und eng auf die Prozessindustrie zurechtgeschnitten, hat die ISO 15926 mittlerweile genügend Semantik, um eine Großzahl an Modellen aus unterschiedlichen Domänen abbilden zu können. Kern ist eine sehr mächtige Ontologie, die ein generisches Datenmodell enthält. Diese ist ein Schnittstellenmodell, welches Anspruch auf Vollständigkeit und Allgemeingültigkeit besitzt. Dies wird durch intensiv durchdachte aufeinander aufbauende Ontologien erreicht.

Als Best Practise kann DEXPI (Data Exchange in the Process Industrie)<sup>4</sup> gelten. DEXPI baut eine Sandbox für ISO 15926, die speziell auf die Prozessindustrie zugeschnitten ist [Fia11] und damit eine Referenzbibliothek für die ISO 15926 darstellt. Hier werden die generische ISO 15926-Konzepte auf konkrete Merkmale der Prozessindustrie spezialisiert, die dann in der Anwendung durch Referenzen von Instanzen genutzt werden können.

Mit Facades und dem PROTEUS 3 Schema gibt es Möglichkeiten zum dynamischen Zugriff, mit dem die Abfrage und der Austausch von Daten ermöglicht wird. Diese sind aber durch fehlende Werkzeuge und Schwäche in der Ausdrucksmächtigkeit nur bedingt geeignet.

Kritik erhält ISO 15926 unter anderem, weil es als schwergewichtig gilt und eine hohe Lernkurve aufweist. Damit erscheint es vor allem für langzeitstabile Informationsmodelle geeignet, deren Qualität durch einen intensiven Ontology-Engineering-Prozess gesichert wurde. Hier ist die Beibehaltung eines akzeptierten Versionsstands eine Qualität, die es unbedingt zu bewahren gilt.

Laut Wiesner verhindert die hohe Komplexität des Datenmodells eine breite Akzeptanz [Wie12, S. 23], weil die Einarbeitungszeit sehr groß ist und

---

<sup>2</sup><http://latc-project.eu/>

<sup>3</sup>[https://docs.google.com/document/d/150dJSMZk5W5ucF23hGj62DaoKtTk9qeaEPBN\\_VCCihI](https://docs.google.com/document/d/150dJSMZk5W5ucF23hGj62DaoKtTk9qeaEPBN_VCCihI)

<sup>4</sup><http://dexpi.org/>

die Möglichkeiten, falsch zu modellieren, groß sind.

## 4.3 OPC UA

OPC Unified Architecture (OPC UA) ist eine Middleware-Lösung mit einem mächtigen Informationsmodell [MLD09]. Das Datenmodell unterstützt Objekt-Orientierung und bietet eine Reihe von Services an, um auf die Daten in verschiedenster Weise zuzugreifen. OPC UA wird von der OPC Foundation entwickelt und gepflegt. Dabei werden die erzielten Ergebnisse zeitnah in die Normenreihe IEC 62541 [IEC] überführt.

Ursprünglich war OPC ein Versuch aus der Automatisierungs- und IT-Domäne einen Kommunikationsstandard für den Shop Floor zu etablieren. Dabei wurde auf OLE als Technologie zurückgegriffen und damit eine Anknüpfung an das Betriebssystem von Microsoft ausgenutzt. Daher stammt auch das ursprüngliche Akronym OPC, das für „OLE for Process Control“ steht. Mittlerweile wird OLE als zugrunde liegende Kommunikationstechnik nicht mehr genutzt, so dass OPC nun für *Openness, Productivity and Collaboration* steht.

Die erste Spezifikation, bekannt als OPC DA (Data Access), verwendete COM zur Kommunikation zwischen einem OPC UA Server und OPC UA Clients und konnte so aktuelle Daten von Items eines Automatisierungssystems verschiedenen Systemen einfach zur Verfügung stellen. Nach Erweiterungen mit verschiedenen Technologien (OPC XML, OPC Alarms and Events, etc.) wurde der Druck größer die Spezifikationen zu vereinheitlichen und den Ballast der zu Grunde liegenden Technologien abzugeben.

Aus diesem Ansatz heraus entstand OPC Unified Architecture (OPC UA), das eine komplette Neuentwicklung darstellt und nun auf offenen Netzwerkverbindungen aufbaut und einem dienst-orientierten Ansatz folgt. Es verfolgt damit den Ansatz, größere Bereiche der Automatisierungspyramide abzudecken, was durch ein mächtiges Informationsmodell [Mah+11] unterstützt wird. Anders als bei OPC DA, bei dem Server, Gruppen und Items als Modellierungselemente vorgegeben waren, erlaubt OPC UA dem Nutzer das benötigte Informationsmodell des Servers selbst flexibel zu gestalten. Dazu kann das Basismodell erweitert um so beliebige Systeme in OPC Servern abzubilden. Diese Flexibilität hat es ermöglicht bereits verschiedene Standards in OPC UA Informationsmodelle zu übertragen.

OPC UA kann somit als Basistechnologie für andere Ansätze dienen, z.B. für FDI (Field Device Integration). OPC UA hat eine zunehmende Verbreitung

in der Prozessindustrie.

Insgesamt zeigt sich OPC UA als ein richtiger Schritt in die Richtung von flexiblen, vernetzten Systemen in der Automatisierungstechnik. Auf Grund seines semantischen und erweiterbaren Informationsmodells lassen sich intelligente, flexible und performante Systeme bauen. Ein Nachteil von OPC UA ist, dass keine gleichberechtigte Integration von mehreren Servern möglich ist. Die Alternative besteht in *Aggregating Server*. Weiterhin ist momentan noch keine echtzeitfähige Kommunikation möglich, was vor allem für die Vernetzung auf den unteren Ebenen der Automatisierungspyramide eine wesentliche Voraussetzung ist. Dies wurde jedoch bereits als Problem erkannt und mit Hilfe der OPC UA Realtime Working Group<sup>5</sup> sowie diversen Forschungsprojekten angegangen.

### 4.4 Eclipse EMF

Das EMF-Projekt<sup>6</sup> ist ein Modellierungs- und Datenintegrationsframework für Eclipse mit Möglichkeit zu Code-Generierung. Damit ist Erstellung von Werkzeugen und Applikationen möglich, die auf einem gemeinsamen und teilbaren strukturierten Datenmodell aufbauen. Modelle werden dabei in XMI<sup>7</sup> beschrieben. EMF bietet für diese Werkzeuge und Laufzeitunterstützung an, um daraus einfach Java-Klassen und XML<sup>8</sup>-Spezifikationen zu erstellen. Desweiteren gibt es einen Editor und Adapter, die den Zugriff und das Verändern des Modells sowohl manuell als auch kommando-orientiert erlauben.

Ecore ist das EMF Modell, welches erlaubt, Modelle zu beschreiben. Dabei ermöglicht es Objektattribute, Assoziationen zwischen Objekten, auf einem Objekt ausführbare Operationen und einfache Einschränkungen (Kardinalitäten etc.) von Objekten zu spezifizieren. Damit bildet es in etwa das Klassendiagramm-Subset von UML ab. Neben der reinen Modellbeschreibung sind mit EMF für Ecore auch Möglichkeiten zur Laufzeitunterstützung möglich. Dies schließt z.B. Änderungsbenachrichtigungen, Persistierungen und eine reichhaltige API zur Manipulation von EMF Objekten ein.

Auf dem gemeinsamen Standard EMF für Datenmodelle haben sich viele Technologien und Frameworks aufgebaut. Darunter gibt es Serverlösungen,

---

<sup>5</sup><http://www.br-automation.com/en/company/press-room/br-supports-opc-foundations-real-time-working-groups/>

<sup>6</sup><http://www.eclipse.org/modeling/emf/>

<sup>7</sup><http://www.omg.org/spec/XMI/>

<sup>8</sup>Extensible Markup Language

Persistierungsframeworks, UI Frameworks und Transformationswerkzeuge, so dass aus den Einzellösungen ein gemeinsamer mächtiger Informationsraum entstehen kann.

## 4.5 AutomationML

Die AutomationML<sup>9</sup> (Automation Markup Language) in Version 2.0 ist ein neutrales XML-basiertes Datenformat für die Modellierung von Planungsinformation in der Automatisierungstechnik [Dra10]. Ziel von AutomationML ist der Austausch von Engineering-Daten in einer heterogenen Tool-Landschaft von modernen Engineering-Werkzeugen für verschiedene Disziplinen wie zum Beispiel Mechanisches Design, Elektrisches Design, HMI-Entwicklung, SPS-Programmierung oder Robotersteuerung"[DC12]. Das Projekt wurde 2006 gestartet und seitdem kontinuierlich mit breiter Industrie-Unterstützung weiterentwickelt, so dass es mittlerweile einen breite Anwendungscommunity gefunden hat und in IEC 62714 [DC12] standardisiert wurde.

Die Kernidee von AutomationML ist die Bereitstellung eines offenen Container-Formats zur Speicherung und zum Austausch von Anlagenplanungsdaten. Dabei wird auf bereits existierende Standards aufgebaut. Technisch wird dazu XML genutzt und auf CAEX [DF04a; DF04b] zur Abbildung der Topologie und Mengengerüste sowie der Attribute und Beziehungen von Objekten in ihrer hierarchischen Anlagenstruktur aufgebaut. *COLLaborative Design Activity* (COLLADA)<sup>10</sup> [AB06] wird zur Darstellung der Geometrie mit detaillierten 3D-Modellen verwendet. Kinematik mit seinen Verbindungen und Abhängigkeiten von Objekten, um Bewegungsplanung zu beschreiben, werden in AutomationML ebenfalls durch COLLADA abgebildet. Schlussendlich bleibt noch PLCOpenXML<sup>11</sup>, um Logik (Ablaufsequenzen, internes Verhalten und I/O-Verbindungen) abzubilden. Durch den Ansatz ist die Erweiterbarkeit von AutomationML, um relevante Inhalte gegeben, die ebenfalls mit XML-basierten Datenaustauschformaten ausgedrückt werden können. So können Best Practises in Companion Standards überführt werden um domänenspezifische Modelle mit AML abzubilden.

Dabei wird CAEX als Dachformat ähnlich einer Ontologie genutzt. Dort sind Typen und Konzepte als sogenannte *SystemUnitClass* definiert, wohingegen Instanzen in einer *InstanceHierarchy* als *InternalElements* abgelegt sind.

---

<sup>9</sup><https://www.automationml.org>

<sup>10</sup><https://www.khronos.org/collada/>

<sup>11</sup>[http://www.plcopen.org/pages/tc6\\_xml/](http://www.plcopen.org/pages/tc6_xml/)

Über *InterfaceClasses* werden Schnittstellen definiert und damit Verknüpfungen zwischen Elementen erstellt. Über *RoleClasses* werden den Elementen spezifische Bedeutungen zugewiesen. Für den Verweis zwischen den einzelnen Teilmodellen können so typisierte Verknüpfungen genutzt werden.

AutomationML erfreut sich wachsender Beliebtheit in der Automatisierungstechnik, so dass eine Reihe von Anwendungen und Werkzeugen dafür entstanden ist. So gibt es Modellierungswerkzeuge, Transformationsansätze, Integration in Programmierumgebungen (z.B. AutomationML Engine<sup>12</sup>), sowie einen OPC UA zu AutomationML Konverter

Nachteile sind die Konzentration auf Abbildung statischer Informationen. Weiterhin ist eine Referenzierung zu anderen Modellen (in entfernten XML-Dateien) zwar möglich, wird in AutomationML aber nicht als Standard betrachtet. Für AutomationML ist eine statische, lokale Datenhaltung der Ausgangspunkt, was eine Konsistenz der Datenmodelle untereinander garantiert.

### 4.6 WBEM/CIM

Getrieben aus dem Bereich in der Informatik hat sich WBEM/CIM als dynamischen Informationsmodell für Standardfunktionen zur Administration und Fernwartung von Computersystemen in Managed Environments durchgesetzt. Dabei steht das WBEM für *Web-Based Enterprise Management* und bezeichnet das Kommunikationsmodell und CIM steht für das dabei eingesetzte Informationsmodell mit dem Namen *Common Information Model*.

Beide Technologien sind unabhängig voneinander, werden jedoch häufig zusammen eingesetzt. Die Spezifikation erfolgt bei beiden durch die Distributed Management Task Force (DMTF).

**WBEM** WBEM<sup>13</sup> ist ein dienst-orientierter Ansatz für das Verwalten von Ressourcen, der häufig für die Verwaltung von IT-Systemen eingesetzt wird. Da er unabhängig von Hardware-Plattform und Betriebssystem ist, hat er in diesem Bereich eine große Verbreitung erreicht.

WBEM unterstützt eine ganze Reihe von Protokollen (z.B. CIM-XML, WS-CIM, CLP-SSH) und Schnittstellen für den Zugriff auf die WBEM-Infrastruktur mittels Internet-Techniken. Dabei verfolgt WBEM einen Client-Server-Ansatz, in dem ein Server mit Hilfe von Providern Informationen über

---

<sup>12</sup><https://github.com/AutomationML/AMLEngine>

<sup>13</sup><http://www.dmtf.org/standards/wbem>

angeschlossene Ressourcen einsammelt und zentral verschiedenen Clients zur Verfügung stellt.

Auf Client-Seite existieren hochsprachliche Zugriffsmethoden auf Managed Objects, so dass auf diese direkt mit Operationen und Events zugegriffen werden kann. Darüberhinaus existieren Möglichkeiten zum Explorieren der Managed Objects sowie komplexe Abfragesprachen. So kann die CIM Query Language für die Abfrage von Informationen aus dem CIM genutzt werden.

**CIM** WBEM wird üblicherweise zusammen mit dem Common Information Model (CIM) <sup>14</sup> verwendet. Dieses Informationsmodell ist objektorientiert, hierarchisch und flexibel. Es unterstützt als Modellierungskonzepte Klassen, Eigenschaften und Methoden. Es kann auch komplexe Relationen zwischen Konzepten abbilden. CIM wird im Managed Object Format (MOF) repräsentiert, was ein UML Dialekt darstellt. Momentan sind in CIM etwa 1400 Klassen definiert, die zur Beschreibung von Managementinformationen und Funktionen in einem Softwaresystem dienen. Diese können unabhängig von der Implementierung (z.B. durch WBEM genutzt) werden und sind ihrerseits durch Vererbung einfach erweiterbar. Die CIM-Infrastruktur enthält ein Meta-Schema und Regeln für die Integration des Zugriffs auf andere Management-Modelle ins CIM, wie beispielsweise DMI, SMBIOS oder SNMP.

**Implementierungen** Es existieren verschiedene WBEM/CIM-Implementierungen. Die bekannteste und am weitesten verbreitete Implementierung ist die Windows Management Instrumentation (WMI), die für Windows eine der wichtigsten Schnittstellen für die Administration und Fernwartung über das Netzwerk darstellt. Damit ist diese bereits auf einer großen Anzahl an Desktop-Rechnern installiert.

## 4.7 Zusammenfassende Bewertung

Tabelle 4.1 stellt die Erfüllung der Anforderungen der einzelnen Technologien zu den Anforderungen aus Kapitel 3.2 zusammen. Dabei erfolgt eine Bewertung der Ansätze, ob sie die Anforderungen vollständig (✓✓), teilweise (✓) oder nicht erfüllen (✗). Die Kriterien für die Bewertung sind dabei im Anhang in den Tabellen A.1, A.2 und A.3 ab Seite 108 detailliert aufbereitet. Die Bewertung bezieht sich dabei auf die Abdeckung durch die Technologien und deren etablierten Erweiterungen und Werkzeuge.

---

<sup>14</sup><http://www.dmtf.org/standards/cim>

**Tabelle 4.1:** Bewertung der Technologien hinsichtlich der aufgestellten Anforderungen

	Anforderung	Semantic Web	OPC UA	WBEM/CIM	ISO 15926	Eclipse EMF	AutomationML
Beschreibungs- m.	B1) Mächtigkeit der Informationsmodellierung	✓✓	✓✓	✓	✓✓	✓✓	✓
	B2) Semantik	✓✓	✓✓	✓	✓✓	✓	✓
	B3) Erweiterbarkeit	✓✓	✓✓	✓	✓	✓	✓✓
	B4) Selbstbeschreibungsfähigkeit	✓✓	✓	✓	✗	✗	✓
	B5) Verknüpfungen	✓✓	✓	✓	✓✓	✓	✓
Methoden	M1) Abstraktion der Kommunikation	✓✓	✓✓	✓	✓✓	✓	✗
	M2) Statische Daten	✓✓	✓	✓✓	✓✓	✓✓	✓✓
	M3) Dynamische Daten	✗	✓✓	✗	✗	✓	✗
	M4) Informationssicherheit	✓	✓	✗	✗	✗	✗
	M5) Dienste-Orientierung	✓✓	✓✓	✓	✗	✗	✗
Werkzeuge	W1) Modelltransformationen	✗	✗	✓	✗	✓✓	✗
	W2) Revisionierung	✗	✗	✗	✓	✓	✗
	W3) Entwicklungsunterstützung	✓✓	✓	✓	✓	✓✓	✓
	W4) Schnittstellen/Adapter für andere Systeme	✓✓	✓	✓	✓	✓	✓
	W5) Inspizierbarkeit zur Laufzeit	✓	✓	✓✓	✓	✓✓	✓



Jeder der vorgestellten Technologien hat einige gravierende Nachteile.

So ist das Semantic Web gut darin, statische Inhalte abzubilden und über Domänen hinweg zur Verfügung zu stellen, doch bei der Behandlung von dynamischen Daten gibt es noch etliche offene Punkte.

OPC UA ist hingegen hoch spezialisiert auf den Austausch von dynamischen Daten mit einem hohen Protokoll-Overhead und schwacher Werkzeugunterstützung bei Modelltransformation und Versionierung.

AutomationML hat seine Stärken in der Beschreibung von statischen Informationen und in der prinzipiellen Erweiterbarkeit durch die Integration bestehender Standards. Da die AML-Spezifikation keinerlei Architektur oder Middleware-Dienste für verteilte AML-Server definiert, existiert keine Lösung zum Austausch von dynamischen Daten. Außerdem bietet es von Haus aus nur schwache Referenzen auf andere Domänen an.

WBEM/CIM ist ein sehr schwer gewichtiger Ansatz, der durch das reichhaltige Informationsmodell zwar eine Menge Einstiegspunkte für Erweiterungen vorhält. Dieses ist jedoch ausgelegt auf IT und erfordert einen hohen Einarbeitungsaufwand.

ISO 15926 ist ebenfalls ein sehr schwergewichtiger Ansatz, der in einem Standardisierungs-Deadlock verharret. Durch die langen Zyklen dauert es lange, bis die Teilmodelle von ISO 15926 einen ausreichend hohen Standardisierungs-Reifegrad nach Drath und Barth [DB12] erreicht haben. In der Zwischenzeit haben sich jedoch die Anforderungen geändert, so dass erneut das Modell nachgearbeitet werden muss.

Insgesamt zeigt sich, dass sowohl das Semantic Web als auch OPC UA einen Großteil der Anforderungen bereits vollständig oder teilweise erfüllen. Linked Data fokussiert dabei stärker auf die Integration heterogener statischer Datenquellen, wohingegen die Stärken von OPC UA in der Verwaltung von dynamischen Daten liegt.



## 5 Konzept - Linked Enterprise Data (LED)

### Aufbau und Einordnung des Kapitels

Dieses Kapitel beschreibt als Kernelement dieser Arbeit *Linked Enterprise Data* als Konzept für die Erstellung und Verwaltung eines interorganisationalen industriellen Informationsraums. Das Konzept erfüllt die gestellten Anforderungen, in dem es auf Linked Data aufbaut und dieses um industrielle Aspekte ergänzt. Der Einsatz von Linked Data in der Industrie hat für diese enorme Vorteile, wie eine einfache Vernetzbarkeit mit anderen Daten und damit einhergehend eine Interoperabilität zwischen Systemen.

Dazu wird zuerst im Abschnitt 5.1 der Beitrag von Linked Data zur Erfüllung der Anforderungen beschrieben. Daraus werden die offenen Punkte abgeleitet und das Konzept im Ganzen vorgestellt und daraufhin auf einzelne Aspekte und deren Anforderungserfüllung eingegangen wird. Dazu wird die Architektur (Abschnitt 5.3), die Bereitstellung von Informationen (Abschnitt 5.4), die Zugriffssicherung von Informationen (Abschnitt 5.5), die Versionierung (Abschnitt 5.6), Modelltransformation (Abschnitt 5.7) sowie auf Möglichkeiten zur flexiblen Standardisierung (Abschnitt 5.8.1) eingegangen.

### 5.1 Abdeckung der Anforderungen durch Linked Data

#### 5.1.1 Erfüllte Anforderungen

Der Semantic Web Stack ist in Abbildung 5.1 zu sehen und deckt mit seinen Technologien bereits einen Großteil der Anforderungen an einen Informationsraum mit den gestellten Anforderungen ab.

Auf der untersten Ebene wird URI/IRI, HTTP, Unicode und Authentisierung in der Web-Plattform-Ebene genutzt. Darauf setzen Datei-Formate (XML, Turtle<sup>1</sup>, RDFa, ...) auf, mit denen eine Serialisierung und somit der Zugriff auf Daten ermöglicht wird. Damit lassen sich bereits die Anforderungen *M1* und *M2* abbilden.

Darauf aufbauend wird RDF als Datenmodell verwendet, in dem alle Informationsmodelle abgebildet werden. Dies erlaubt die Erfüllung der Anforderung

---

<sup>1</sup>Terse RDF Triple Language [BB11]

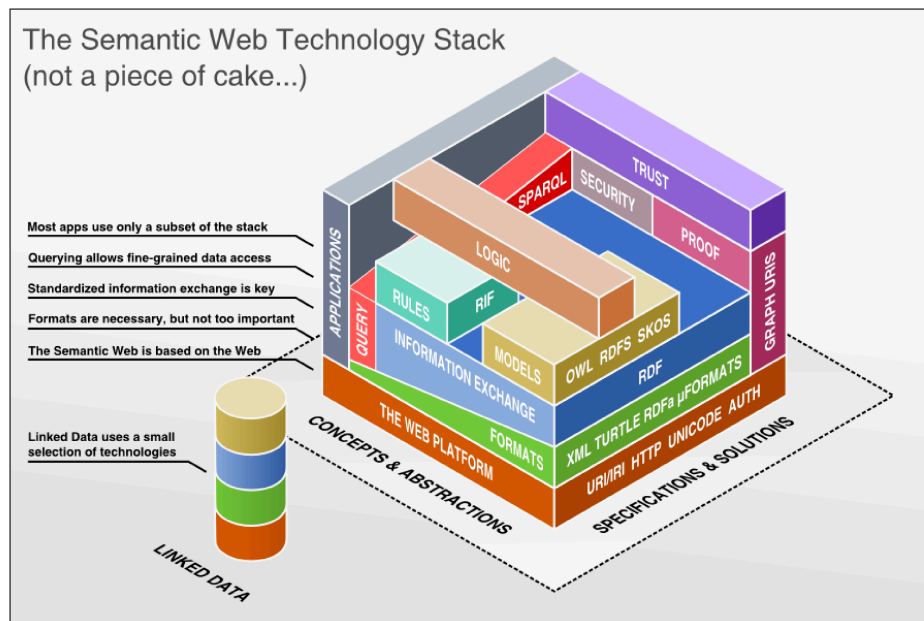


Abbildung 5.1: Semantic Web Stack [zaz11b]

derungen  $B3$  und  $B5$ , da RDF durch seine Triple-Struktur einfach erweiterbar ist und durch die Verwendung von URIs auch organisationsübergreifende Verknüpfungen zum Standardumfang gehören.

Der ursprünglich rein auf HTTP basierende Zugriff auf Linked Data hat durch die Ergänzung mit der Abfragesprache SPARQL einen echten Mehrwert erhalten. Mit diesem ist es nun möglich, komplexe Informationszusammenhänge direkt vom Server auswerten zu lassen und so die Clients zu entlasten, die sich sonst eine große Menge an Daten zuerst herunterladen müssten um darauf ihren Zugriff direkt durchzuführen. Weitere Dienste zum Zugriff auf RDF Informationen werden durch HTTP-Requests abgebildet, die sich üblicherweise in einer REST<sup>2</sup>-Schnittstelle ausdrücken, so dass auch Anforderung  $M5$  erfüllt ist. Berners-Lee [BO13] stellt als eine weitere wichtige Anforderung an die Weiterentwicklung von Linked Data heraus, dass neben dem gut entwickelten Lesezugriffen der Schreibzugriff vereinfacht und somit häufiger genutzt wird. Insofern ist hier in den nächsten Jahren auch mit weiteren Fortschritten zu rechnen.

Mit Hilfe von der Semantik in den genutzten Informationsmodellen (OWL, RDFS, ...) können die Modelle so angereichert werden, dass diese sich aus-

<sup>2</sup>Representational State Transfer

drucksstark und maschineninterpretierbar selbst beschreiben können. Durch Reasoner und höherwertiger Regeln (RIF) können komplexe Logiksysteme aufgebaut werden ( $B1$ ,  $B2$ ,  $B4$ ).

In der Einordnung in den ISO/OSI-Stack mit seinen sieben Schichten [II94] ist der Semantic Web Stack ab Ebene 4 zu finden. LED setzt auf den normalen Internet-Stack mit TCP/IP auf. Es definiert größtenteils den Informationsaustausch in der Anwendungsschicht (Applicationlayer auf Schicht 7). Es ist prinzipiell zwar möglich, die unterliegenden Schichten auszutauschen. Damit verliert das System jedoch die Interoperabilität mit anderen Werkzeugen, was eine der großen Stärken von LD ist. Daher ist LD grundsätzlich in der Leistungsfähigkeit eingeschränkt. Eine voll-kompatible Architektur ohne aufwändiges TCP ist nicht realisierbar.

In dem Bereich der Werkzeuge werden durch die große Verbreitung in der IT ebenfalls bereits Anforderungen erfüllt. Für das Semantic Web steht eine große Anzahl an Werkzeugen zur Verfügung, mit denen sich Arbeiten im ganzen Lebenszyklus von Daten unterstützen lassen. Viele der Werkzeuge sind als Open-Source frei verfügbar und werden durch eine aktive Community getrieben. Desweiteren gibt es durch die einsetzende Verwendung in geschäftlichen Anwendungen auch eine Reihe von Tools für die Firmen professionellen Support liefern. Eine Auswahl an wichtigen Werkzeugen ist im Anhang im Abschnitt C.2 zusammengefasst, die sowohl Applikationsentwicklung unterstützen  $W3$  als auch eine Vielzahl an Schnittstellen/Adapter für andere Systeme  $W4$  bereitstellen.

### 5.1.2 Notwendige Erweiterung zu LED für einen industriellen Informationsraum

Wie in Kapitel 4 beschrieben, erfüllt der Ansatz des Semantic Web einen Großteil der aufgestellten Anforderungen. Es bleiben jedoch folgende Punkte offen:

**Anbindung an existierende industrielle Informationssysteme (W4)** Für die Befüllung eines Informationsraums sind die entsprechenden LD-Adapter notwendig, die die Informationen aus den existierenden Werkzeugen holen und in diese zurückschreiben. Dies ist sowohl bei einer kompletten Ablösung der alten Systeme als auch bei einer fortschreitenden Synchronisation mit diesen Systemen notwendig.

**Performante Einbindung dynamischer Daten (M3)** Die Vielzahl an Sensoren und Aktoren müssen in den LED-Raum eingebunden werden. Hier haben sich in der näheren Vergangenheit einige Technologien herausgebildet, mit denen ein einfacher und schneller Zugriff auf schnell ändernde Daten in komplexen Zusammenhängen möglich ist (MQTT, DDS, OPC UA). In dieser Arbeit wird sich dabei auf die Integration der weit verbreiteten OPC UA Technologie konzentriert, da sie von diesen in der betrachteten Domäne momentan die größte Verbreitung aufweist.

**Integrierte Mechanismen für Modelltransformation (W1)** Modelltransformationen sind kein Kernbestandteil von Linked Data, können aber durch die Mächtigkeit von SPARQL ausgedrückt werden. Dazu sind jedoch entsprechende Mechanismen und Regeln aufzustellen.

**Revisionierbarkeit (W2)** Revisionierbarkeit spielt bei Linked Data kaum eine Rolle. Es gibt wenige proprietäre Lösungen. Für eine langfristige und stabile Revisionierungslösungen sind diese jedoch mit dem semantischen Potential von Linked Data zu erweitern.

**Unterstützung bei der Nachverfolgbarkeit (W5)** Gerade bei großen und verteilten Informationsräumen kann der Mensch schnell die Übersicht über die Daten verlieren. Hier ist dem Nutzer durch geeignete Visualisierungssysteme Hilfestellung unter Einbeziehung bekannter Visualisierungskonzepte zu leisten.

**Feingranulare Informationssicherheit (M4)** Die Informationsabsicherung von dereferenzierbaren URIs durch Authentifizierungs- und Autorisierungsmechanismen ist sehr aufwändig. Dies kann prinzipiell durch ein geschicktes HTTP Redirecting auf einen SPARQL Endpoint gestaltet werden, der die zugreifbaren Informationen über ein SPARQL CONSTRUCT bereitstellt. Somit kann das Problem auf die Zugriffssicherung von SPARQL zurückgeführt werden.

## 5.2 Kernkonzept von Linked Enterprise Data

Bemühungen einen integrierten Informationsraum zu schaffen, gibt es bereits lange, da die Vorteile klar auf der Hand liegen. So wurde z.B. in den 90er mit CIM (Computer-integrated manufacturing) ein Ansatz zur Vollintegration

aller Informationen und Dienste gestartet, der aus der Produktion getrieben wurde. Der Ansatz ist jedoch an der fehlenden Agilität und großen Starrheit zu Grunde gegangen. Vertreter waren STEP [DI04] und das daraus folgende ISO 15926 [ISO07]. In den 00er wurde dann aus der Informatik der Ansatz des Semantic Web gestartet, der explizit auf eine volle Flexibilität gesetzt hat. Dieser konnte sich aber zumindest in konservativen Industrien auf Grund der Praxisferne nicht durchsetzen. Heute steht nun eine Zusammenführung der beiden Ansätze für eine flexible Produktion für Industrie 4.0 an, die einen notwendigen Ausgleich zwischen Flexibilität und Stabilität schafft. Diesen Ausgleich versucht *Linked Enterprise Data* zu schaffen, in dem es *Linked Data* als Ausgangspunkt nimmt und um entsprechende Methoden und Konzepte in den Bereichen Werkzeuge und Methoden für eine industrielle Nutzung erweitert.

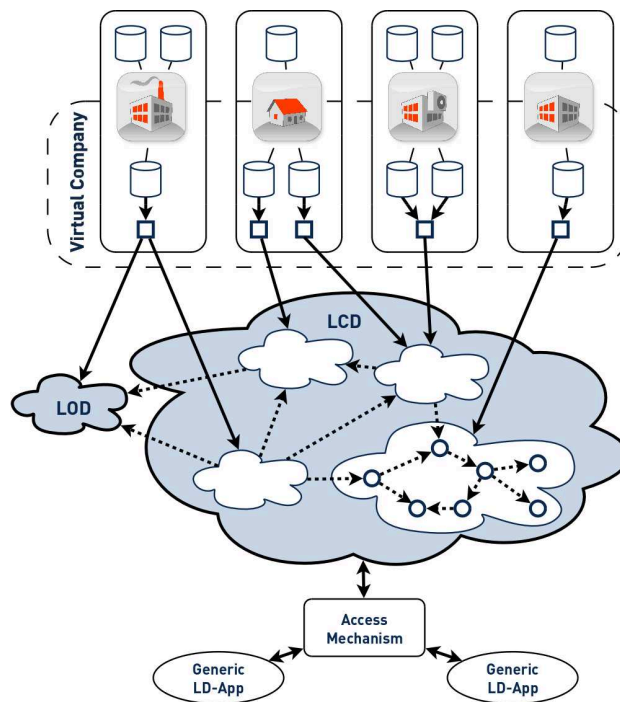
Diese Arbeit versucht diese Aufgabe durch das Konzept *Linked Enterprise Data* zu lösen. Der Begriff selbst wurde bereits in [Gra+12] eingeführt und wird hier nun als ein *auf dem Semantic Web aufbauender und für die Anforderungen der Industrie erweiterter integrierter Informationsraum* definiert. Das Kernkonzept eines *Linked Enterprise Data Informationsraums*, wie er in dieser Dissertation verstanden wird, ist in Abbildung 5.2 zu sehen. Dabei fungiert der LED-Cloud als gemeinsamer Informationsraum, in den verschiedene Applikationen ihre Daten spiegeln und semantisch auf Datenebene integrieren können [Gra+11; Gra+12].

Es besteht somit aus einem verteilten Data-Warehouse, das die Informationen auf Datenebene integriert. Dabei werden die Warehouses jedoch nicht zwangsläufig synchronisiert, sondern beinhalten verschiedene Datensätze. Eine Synchronisation ist aber trotzdem möglich. Mittlerweile hat sich SPARQL weit verbreitet [Rak+13], so dass der Hauptzugriff auf Informationen in einem LED-Netzwerk über SPARQL stattfindet.

*Linked Enterprise Data* (LED) hat einige Besonderheiten im Vergleich mit *Linked Open Data* (LOD). *Linked Enterprise Data* beinhaltet auch Informationen, die nicht frei für jedermann verfügbar sein sollen. Zudem spielen hier ganz entscheidend dynamische Schreibprozesse in die LED-Cloud eine Rolle.

Damit sind folgende Vorteile erreichbar:

Es ist kein Weltmodell notwendig, das alle Aspekte jeder Anwendung beschreibt. Dieses Ziel ist unrealistisch, da sich die zu beschreibende Welt ständig weiterentwickelt und so dieses große Modell permanent von den Entwicklungen in der Realität überholt wird.



**Abbildung 5.2:** Konzept in einer Nussschale



Der Ansatz setzt auf vorhandene spezialisierte Werkzeuge auf, so dass für die Nutzer keine Änderung auftritt und auch die Werkzeuge selbst nicht geändert werden müssen, solange sie geeignete Schnittstellen für den Import und Export von Informationen bereitstellen.

Es sind mächtige Abfragen durch *Semantic Lifting* möglich, die gleichzeitig mehrere Datenquellen in ihre Abfragen einbeziehen können (Federated Queries)

Die Systeme wachsen mit den Anforderungen, da einerseits neue Konzepte einfach hinzugefügt werden können und andererseits Änderungen an den Modellen ebenso einfach möglich sind und mit einem Versionsverwaltungssystem für die Nutzer nachvollziehbar bleiben und gleichzeitig rückwärts-kompatibel gehalten werden können.

## 5.3 Dezentrale Architektur

### 5.3.1 LED-Architektur

Ein LED-System muss verteilt sein, um den Ansprüchen eines organisationsübergreifenden Informationsraums zu genügen. Nur so hat jeder Partner die Möglichkeit, Herr über seine Daten zu bleiben und den Zugriff auf relevante Informationen unter seiner Kontrolle zu behalten. Für kleine Informationsräume mit wenigen Partnern ist eine Verteilung der notwendigen Komponenten bei einem Partner dennoch möglich, wenn die anderen Partner diesem das dafür notwendige Vertrauen entgegen bringen und durch entsprechende Service Level Agreements absichern. Dabei können alle Partner Informationen bereitstellen, die sie entweder direkt in Linked Data angelegt haben oder aus ihren bereits vorhandenen Informationssystemen nach Linked Data überführen. Die Einigung auf Linked Enterprise Data löst dabei alle Fragen bezüglich Syntax. Dies ermöglicht eine Integration auf Datenebene, so dass alle Partner über dieselben Zugriffswege sowohl auf die eigenen als auch die fremden Informationen zugreifen können. Daher muss jeder Partner einen Zugriffskanal auf seine Daten (über SPARQL) bereitstellen, der jedoch zusätzlich über domänenübergreifende Authentifizierungs- und Autorisierungsmechanismen verfügt. Die spezielle Ausgestaltung einer Architektur findet sich in Kapitel 6.1.

### 5.3.2 Verteilte Abfragen (Federated Queries)

Eine der großen Vorteile von Linked Data ist die Einbindung von externen Informationen in den eigenen Informationsraum. So müssen die gewünschten Informationsteile nicht auf die eigenen Server repliziert werden, sondern können auch von anderen Servern genutzt werden, sofern die organisatorischen Voraussetzungen dafür erfüllt sind. Viele interessante Fragen lassen sich jedoch nur unter Einbeziehung von Informationen aus Quellen beantworten, die in dem betreffenden Datensatz verlinkt sind und sich auf einem anderen Server befinden. Wenn diese Daten nun auf unterschiedlichen Servern liegen, muss eine Zusammenführung erfolgen.

Der einfachste Fall besteht darin, dass der Client die Zusammenführung selbst durchführt. Dazu ruft er die Informationen von beiden Datenquellen ab und integriert die Ergebnisse dann lokal. Für den Abruf der Informationen können dabei einfache HTTP-GET-Abfragen oder auch komplexere SPARQL-Queries genutzt werden. Dies bedeutet für den Client jedoch, dass er wissen muss, wo der verlinkte Datensatz liegt, wie er diesen abrufen und wie genau die Integration dann stattfinden soll. Insbesondere muss er also eine Kenntnis darüber haben welche Informationsentität mit welcher in Zusammenhang steht und wie die Semantik der Integration auszusehen hat.

Eine andere Möglichkeit stellen sogenannte *Federated Queries* dar, die eine automatische Zusammenführung der Informationen von mehreren Servern erlauben. Rakhmawati et. al. [Rak+13] haben dafür mögliche Prinzipien und existierende Werkzeuge analysiert und zusammengefasst. Hier hat SPARQL 1.1 die Möglichkeit des SERVICE-Keywords eingeführt. Mit diesem ist es möglich in einer SPARQL-Query eine unterlagerte Query an einen anderen Server zu spezifizieren. Die Ergebnisse aus diesen zusätzlichen Quellen werden direkt als Zwischenergebnis des SPARQL-Endpoints integriert und können so für die weitere Abarbeitung der Query verwendet werden. Insbesondere können bei darüber hinausgehende Mechanismen wie z.B. dem Sponging-Mechanismus von Virtuoso<sup>3</sup> auch noch die abzufragenden Quellen in einer SPARQL-Abfrage ermittelt werden, die in der gleichen Abfrage dann abgerufen und die entsprechende Query darauf angewendet sowie die Ergebnisse aggregiert werden. Eine beispielhafte Query, die Informationen über Arnold Schwarzenegger aus der DBpedia und zu seinen Filmen aus LinkedMDB abrufen, ist in Listing 5.1 zu sehen.

Darüber hinaus gibt es weitere spezialisierte Tools, die einen transparenten Zugriff auf Informationen aus verschiedenen Triplestores ermöglichen ohne

---

<sup>3</sup>[http://uriburner.com/sponger\\_architecture.vsp](http://uriburner.com/sponger_architecture.vsp)

**Listing 5.1:** Federated SPARQL Query

---

```
# SPARQL endpoint queries two additional endpoints and
# integrates the results
# Adapted example from http://www.snee.com/bobdc.blog
# /2010/01/federated-sparql-queries.html

PREFIX imdb: <http://data.linkedmdb.org/resource/movie/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX dbpo: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?birthDate ?spouseName ?movieTitle ?movieDate
WHERE {
  SERVICE <http://dbpedia.org/sparql> {
    ?actor rdfs:label "Arnold Schwarzenegger"@en ;
    dbpo:birthDate ?birthDate ;
    dbpo:spouse ?spouseURI .
    ?spouseURI rdfs:label ?spouseName .
    FILTER ( lang(?spouseName) = "en" )
  }
  SERVICE <http://data.linkedmdb.org/sparql> {
    ?actor imdb:actor_name "Arnold Schwarzenegger".
    ?movie imdb:actor ?actor ;
    dcterms:title ?movieTitle ;
    dcterms:date ?movieDate .
  }
}
```

---

die Notwendigkeit zu wissen, dass sich die Informationen auf verschiedenen Endpunkten befinden. Hier war DARQ<sup>4</sup> ein erster Ansatz, um eine transparenten Abfragemöglichkeit auf verschiedene, verteilte SPARQL-Endpoints zu ermöglichen, die sich wie eine Abfrage auf einen einzigen RDF-Graph verhält. Mit FedX<sup>5</sup> hat fluid Operations eine weitere Möglichkeit vorgestellt. Ebenso existieren eine Reihe weiterer Open-Source-Tools wie SPLENDID<sup>6</sup> und ANAPSID<sup>7</sup>. Letztere ist eine adaptive Abfrage-Engine für SPARQL Endpoints und kann damit Abfragen gezielt verteilen. Hibiscus stellt dafür eine Erweiterung dar, um eine geschickte Aufteilung der Pattern auf verschiedene Endpunkte zu ermöglichen [SN14]. Eine detaillierte Auflistung findet sich bei [Rak+13].

### 5.4 Informationsbereitstellung - LD Adapter

Die Nutzung von vernetztem semantischen Informationsraum setzt voraus, dass die Daten aus den einzelnen Quellen in einen solchen Informationsraum überführt werden. Dafür müssen die Daten jedoch zuerst semantisch angehoben und in RDF abgebildet werden. Dazu ist einerseits Domänenwissen notwendig, das in Ontologien abgebildet wird, andererseits auch Werkzeuge zur Transformation solcher Daten. Für den Zugriff auf die Informationen ist es notwendig, zwischen dynamischen Daten und quasi-statischen Daten zu unterscheiden.

#### 5.4.1 Dynamische Daten

Der Fokus von Linked Data liegt auf der Bereitstellung und Verknüpfung von heterogenen und verteilten Informationsnetzwerken. Bei der Entwicklung der LD-Konzepte hatten die Ersteller im Besonderen statische Daten im Blick. Daher ist die Unterstützung von sich schnell ändernden Daten nur schwach ausgeprägt.

So sind Triple-Stores nicht darauf ausgelegt hoch-dynamische Daten vorzuhalten. Über die üblichen Mechanismen lassen sich keine Updates von mehreren Items in Sekundenrhythmus oder noch schneller durchführen, da dies sowohl Netzwerk als auch Server überlastet. Stattdessen müssen diese Daten on-the-fly in LD gewandelt werden.

---

<sup>4</sup><http://darq.sourceforge.net/>

<sup>5</sup><http://de.slideshare.net/aschwarte/fedx-for-federated-query-processing-on-linked-data>

<sup>6</sup><http://de.slideshare.net/OlafGoerlitz/splendid-9858478>

<sup>7</sup><https://github.com/anapsid/anapsid>

Die Integration von dynamischen Daten in die Daten aus der Planungsphase, die üblicherweise bereits im LED vorliegen, ergibt eine Reihe von Vorteilen. So sind mächtige Abfragen möglich, die über beide Datenarten durchgeführt werden. Für andere Systeme gibt es keine vorgefertigten Adapter, aber es hat sich mit der LDP ein Standard/Richtlinie zum Erstellen solcher Adapter herausgestellt.

Prinzipiell ist durch die Nutzung von HTTP eine relativ hohe Latenz bei der Datenabfrage gegeben, da die Verbindung jedes Mal komplett neu aufgebaut werden muss. Hier gibt es aber Forschungen im Bereich von Linked Data Streams [LPH12; SC09].

### 5.4.1.1 Linked Data Plattform (LDP)

Die Linked Data Plattform<sup>8</sup> (LDP) ist eine Richtlinie zum Bereitstellen von dynamischen Daten in Linked Data Umgebungen. Als Schnittstelle dienen REST-Services, die sich durch ihre Eigenschaften gut in die Welt des Semantic Web integrieren lassen [PDM11]. Beide haben Ressourcen als Grundlage, die über URIs bzw. IRIs identifiziert werden. Damit lassen sich alle Ressourcen in Linked Data und REST eindeutig identifizieren. Dies ermöglicht einen fundamentalen Wechsel in der Betrachtungsebene. Früher wurden nur Geräte als eigenständige Ressourcen behandelt und informationstechnisch dargestellt. Dies wurde mittlerweile auf einzelne Objekte auf die Geräte ausgeweitet, die nun selber identifiziert werden können. Nun können aber auch die einzelnen Informationsentitäten der Objekte eindeutig identifiziert werden (also z.B. die Liste der möglichen Einheiten für einen Parameter ist selbst eine Informationseinheit, an die weitere Informationen angehängt werden können). Während klassische Webservices ihre Ergebnisse häufig in JSON (JavaScript Object Notation) zurückgeben, ist für Linked Data Services eine Serialisierung in einem RDF-Format geboten, wobei hier mit JSON-LD<sup>9</sup> auch eine JSON-konforme Serialisierung zur Verfügung steht, die Webclients einfach parsen können.

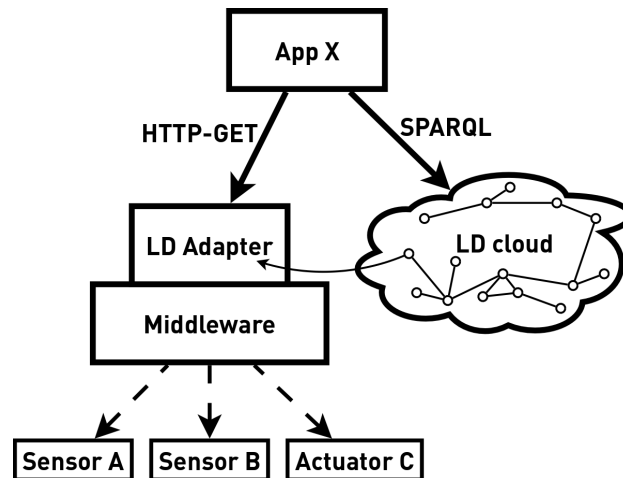
### 5.4.1.2 OPC UA Adapter

Eine Anbindung von OPC UA in das LD Universum gestaltet sich als relativ einfach, da beide viele Konzepte teilen. Dazu gehören ein semantisches Informationsmodell, eine dienst-orientierte Architektur sowie die Nutzung von

---

<sup>8</sup>[http://www.w3.org/2012/ldp/wiki/Main\\_Page](http://www.w3.org/2012/ldp/wiki/Main_Page)

<sup>9</sup><http://www.json-ld.org>



**Abbildung 5.3:** Konzept für den Zugriff auf dynamische Daten

Web-Services.

Für einen nahtlosen Zugriff aus der Linked Data Welt nach OPC UA ist einerseits die Zugriffsmethode semantisch zu beschreiben, um auf die OPC UA Nodes zuzugreifen. Andererseits ist der konkrete Mechanismus zum Zugriff auszugestalten. Die Abbildung 5.3 zeigt dabei, dass ein Client sich zuerst die statischen Informationen zu einem Sensor aus der LD Cloud z.B. per SPARQL abfragt. In diesen ist eine Verknüpfung zu einem LD-Adapter vorhanden, der den Zugriff auf den aktuellen Wert oder auch auf die historischen Werte dieses Sensors bereitstellt. Dieser Zugriff erfolgt dann per REST über eine HTTP Anfrage.

Bei der speziellen Betrachtung von OPC UA als Middleware, verdient es hier jeder OPC UA Knoten, als eigene Ressource behandelt zu werden und damit auch eine REST URI zu erhalten. Damit setzt sich der Weg zu einer feingranulareren Identifikation in Informationssystemen fort. Es ist vorteilhaft, die Möglichkeit zum Verweis auf einen Knoten bereitzustellen, auch wenn die momentane Nutzung für diesen Knoten noch nicht sinnvoll erscheint. Aus Gründen der Weiterentwicklungsmöglichkeiten der Anwendung sollte man sich hier aber diese Option offen lassen. Ein Backlink von OPC UA auf die Linked Data URI kann durch eine Erweiterung der Basisklassen erreicht werden, die dann ein weiteres Attribut für eine Linked Data URI enthalten.

Eine Alternative zu diesem Ansatz wurde z.B. von [GPP15] aufgezeigt. Dieser stellt OPC UA Dienste über Web Services zur Verfügung, in dem direkt im OPC UA Stack das Encoding und Transport-Layer angepasst werden.

Damit ist dann auch eine Kommunikation über UDP möglich. Der hier vorgestellte Ansatz hat aber den Vorteil, dass der Adapter deutlich mehr Informationen in die Antwort stecken kann und diese damit semantisch aufwerten kann.

### 5.4.2 Quasi-Statistische Daten

Neben dynamischen Daten, die sich im Sekundenmaßstab ändern, gibt es auch Daten, die weitaus seltener aktualisiert werden. Diese können daher anders behandelt werden. Die Aktualisierungen dieser Daten findet dabei üblicherweise eher in dem Bereich von Tagen und Wochen statt und werden über manuelle Prozesse ausgelöst. Insgesamt kommt diese Art der Daten eher in der Planungsphase einer Anlage vor. Im Folgenden werden einige Methoden und Werkzeuge vorgestellt, um diese Daten, die in einer beliebigen Form vorliegen, in das LED-Netzwerk zu integrieren.

Dafür stellt die Linked Open Data Community eine Reihe von dynamischen Adaptern bereit für häufig anzutreffende Systeme in diesem Umfeld. Dazu gehören als beispielhafte Vertreter XLWrap und D2RQ.

#### 5.4.2.1 XIWrap

XIWrap<sup>10</sup> ist ein Werkzeug, mit dem sich leicht Tabellen in RDF übersetzen lassen. Die Übersetzung nutzt eine Mapping-Spezifikation, die als RDF Modell vorliegt, vielfältige Filter und Operationen zur Verfügung stellt und somit komplexe RDF Modelle erzeugen kann. XIWrap unterstützt die Transformationen von Microsoft Excel und OpenDocument Tabellen sowie Dateien mit komma- oder tabulator-getrennten Werten (CSV). Die Dateien können sowohl lokal eingelesen als auch von entfernten Servern heruntergeladen werden. Die Bereitstellung als RDF erfolgt in Dateien oder als Web-Service mit SPARQL-Schnittstelle. XIWrap steht unter der Mozilla Public License (MPL) zur freien Verfügung.

#### 5.4.2.2 D2RQ

Die D2RQ<sup>11</sup> Plattform erlaubt es auf relationale Datenbanken als virtuelle RDF Graphen zuzugreifen. Dazu nutzt der D2R Server ein anpassbares

---

<sup>10</sup> <http://xlwrap.sourceforge.net/>

<sup>11</sup> <http://d2rq.org>

Mapping um die tabellenbasierten Daten auf RDF abzubilden und diese auf einem SPARQL-Endpoint sowie Browser zur Verfügung zu stellen.

SPARQL- und Browsing-Anfragen werden in SQL-Abfragen unter Anwendung des spezifizierten Mappings umgeschrieben. Diese on-the-fly Übersetzung erlaubt es, große Datensätze live als Linked Data zur Verfügung zu stellen und benötigt keine Replizierung in einem Triplestore. D2RQ steht unter der Apache Lizenz zur freien Verfügung und der Quellcode wird auf GitHub unter <https://github.com/d2rq/d2rq> bereitgestellt.

### 5.4.2.3 Legacy-Daten

Quasi-statische Daten ändern sich so selten, dass die Entwicklung einer tiefen Integration inklusive einer automatisierten Synchronisation von den Legacy-Werkzeugen in die Linked Data Cloud sich nicht lohnt. Für die Informationen aus den Planungsphasen der digitalen Anlage wird häufig ein CAE<sup>12</sup>-System (z.B. Comos) eingesetzt. Diese haben in der Regel Schnittstellen, die sowohl einen Import als auch einen Export der Daten erlauben. Bei Comos ist dies z.B. über eine frei-definierbare XML-Schnittstelle möglich. Über diese können die Daten exportiert werden, mit Domänenwissen semantisch angereichert und als RDF-Dateien abgelegt werden. Damit ist eine manuelle Synchronisierung in Triplestores ein gangbarer Weg, der jedoch für die einzelnen Werkzeuge und deren Schnittstellen geprüft und gestaltet werden muss.

## 5.5 Zugriffssicherung für Linked Enterprise Data

Unternehmen werden ihre Daten nur dann für andere Partner bereit stellen, wenn sie sich sicher sein können, dass der Zugriff auf die Daten fein granular und sicher gestaltet werden kann. Auf keinen Fall sollen andere Personen außer den Partnern, die über Service Level Agreements (SLA) vertraglich in Haftung genommen werden können, auf diese Daten zugreifen. Zudem soll jeweils auch nur ein gewisser Teil der eigenen Daten überhaupt für diese Kollaborationen genutzt werden. Davon sollen wiederum nur gewisse Teile für bestimmte Partner verfügbar sein. Dies entlastet zeitgleich die Empfänger der Daten, da sie nur für sie relevante Informationen bekommen und somit nicht die gleiche Verantwortung tragen müssten wie in dem Fall, in dem sie einen ungefilterten Datensatz mit potentiellen Geschäftsgeheimnissen erhalten. Dabei soll der Zugriff auf verschiedenen Ebenen kontrolliert werden:

---

<sup>12</sup>Computer Aided Engineering

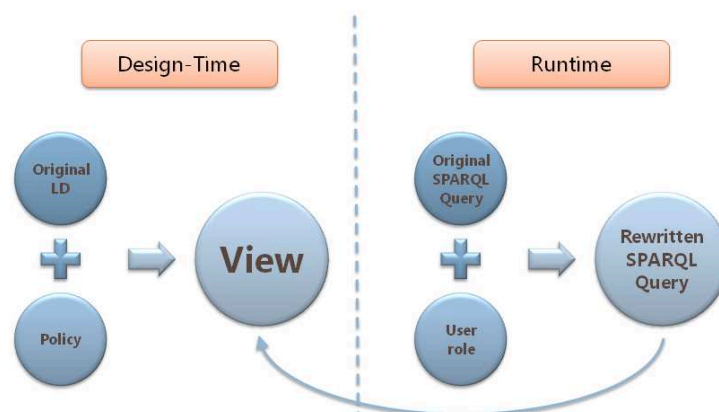


**Datenbank-Ebene** Der Zugriff wird komplett auf einen Triplestore geblockt oder freigegeben. Dieser Filtermechanismus ist relativ einfach durch Netzwerkeinstellungen umsetzbar, bietet jedoch keinen fein-granularen Zugriff.

**Graph-Ebene** Einzelne Named-Graphs in einem Triplestore können freigegeben werden. Einzelne Triplestores (z.B. Virtuoso und Stardog) unterstützen diese Form der Filterung unter Nutzung einer internen Nutzerdatenbank.

**Ressourcen-Ebene** Einzelne RDF Ressourcen (URIs) mit all ihren Properties werden freigegeben.

**Statement-Ebene** Einzelne RDF Aussagen (Triples) können freigegeben werden.



**Abbildung 5.4:** SAPRQL security [Ort+13]

Zusammen mit den Partnern aus ComVantage wurde in [Ort+13] ein Konzept zur Absicherung von LED-Informationen über eine SPARQL Schnittstelle vorgestellt. Diese setzt auf einem gewöhnlichen Triplestore mit SPARQL-Endpoint auf. Dazu werden die Triples in Named Graphs aufgeteilt, die jeweils verschiedene Sichten (Views) auf den Datensatz darstellen. Nun kann jedem externen Partner ein Zugriff auf eine beliebige Anzahl an Sichten ermöglicht werden.

Wie in Abbildung 5.4 zu sehen ist, kann nun ein Client eine normale Abfrage an dem gesicherten SPARQL-Endpoint stellen. Dieser nutzt jedoch

die aus der Authentisierung gewonnenen Nutzerinformationen, um die Query so umzuschreiben, dass die Query nur auf die Informationen aus den erlaubten Views zugreifen darf. Die so umgeschriebene Query wird nun auf den normalen SPARQL-Endpunkt ausgeführt, der an den Triplestore mit den Daten angeschlossen ist. Dieses SPARQL Rewriting erfüllt somit die Aufgabe der Autorisierung für einzelne Views, wobei es diese Informationen direkt aus der Datenbasis nimmt. Hierbei ist es natürlich wichtig, dass die Abfragen von Nutzern nicht direkt auf dem ungeschützten SPARQL-Endpunkt ausgeführt werden dürfen. Weiterhin ist das Schreiben oder Verändern der Named Graphs, die die Zugriffsrechte auf die Views enthalten, separat abzusichern. Details der Umsetzung finden sich in Kapitel 6.4.

### 5.6 Versionierung

Auch quasi-statische Informationen ändern sich im Lauf der Zeit. In Informationssystemen zur Planung von industriellen Projekten ist dies im Normalfall sogar unausweichlich. Diese Änderungen müssen jedoch später noch genau nachvollzogen werden können. Weiterhin muss eventuell auf ältere Versionen zurückgegriffen werden, um bestimmte Entwicklungen nachvollziehen oder korrigieren zu können. Daraus ergibt sich die Notwendigkeit zur Versionierung dieser Informationen.

Dies muss natürlich in das Gesamtsystem tief eingebunden werden können. Andererseits muss die Kopplung so lose sein, dass nicht alle Daten versioniert werden müssen. Dies ist für sich häufig ändernde Daten unter Umständen eine Überbelastung der Speicherfähigkeit des Systems.

Ähnlich wie bei der Zugriffssicherung stellt sich bei der Versionierung die Frage, auf welcher Ebene versioniert werden soll. Eine Revisionierung auf Tripel-Ebene ist hier nicht möglich, da ein Tripel selbst keine Beziehung zu anderen Tripeln hat, sondern lediglich existieren kann oder nicht. Somit geht der Zusammenhang, der durch eine festgelegte Änderung existiert, wieder verloren. Eine Revisionierung auf Ressourcen-Ebene ist hingegen denkbar und wird beispielsweise auch durch das Changeset-Vokabular<sup>13</sup> propagiert. Der Nachteil liegt aber darin, dass man häufig nicht nur Abfragen auf eine Ressource durchführt, sondern auf eine Verknüpfung von Ressourcen aus dem gleichen Graphen. Dabei wird es kompliziert, eine gemeinsame Revision dieser Verknüpfungen zu spezifizieren. Als letzte und hier favorisierte Möglichkeit bleibt noch die Revisionierung auf Graphen-Ebene, die zudem die stärkste

---

<sup>13</sup><http://vocab.org/changeset/schema.html>

**Listing 5.2:** R43ples Select Query

---

```
SELECT *  
FROM <test> #REVISION "23"  
WHERE { ?s ?p ?o }
```

---

Trennung der aktuellen Revision von vorherigen erlaubt und somit ein direktes Arbeiten ohne weitere Einflüsse auf dieser ermöglicht.

Eine Möglichkeit zur ressourcensparenden Speicherung von Revisionen liegt in einer Delta-Kompression, bei denen nur die Unterschiede abgelegt werden. Hierfür bieten sich bei LED Named Graphs in einem Triplestore an, die über entsprechende Verknüpfungen einen Revisionsgraphen bilden, aus dem dann eine beliebige Revision wieder hergestellt werden kann.

Ein Zugriff auf alte Revisionen sollte dabei einfach möglich sein. Idealerweise fügt sich die Schnittstelle in bereits vorhandene Schnittstellen ein. So sollte z.B. eine SPARQL-Query nur einfach erweitert werden müssen, um diese statt auf der aktuellen Revision eines Graphen auf einer anderen Revision auszuführen. Eine Möglichkeit zeigt dazu Listing 5.2, welche ein neues Keyword **REVISION** zur Spezifikation der Revision 23 für den Graphen **test** einbringt. Dabei kann eine solche Revisionsinformation auch an andere Graphen angehängt werden und nicht nur als Revisionsnummer, sondern auch als Bezeichnung eines Branches oder Tags angegeben werden.

Weitergehende Informationen zu dem R43ples-Konzept für eine Revisionsverwaltung mit Zugriff über SPARQL finden sich in [GHU14] und wurden in [GHU16] verfeinert.

## 5.7 Modell-Transformation

Modell-Transformationen treten in unterschiedlichen Phasen und Anwendungsfällen in den meisten Projekten auf [Ehr+06]. Sie sind häufig notwendig, um die in der Planung gewonnenen Informationen in ein Modell zu überführen, das auch im Betrieb sinnvoll genutzt werden kann. Somit können sie zu einer Optimierung der laufenden Kosten (OPEX) beitragen. Wie Rahm in seiner Diplomarbeit zeigt, können sie ein wichtiger Bestandteil zum Round-Trip Engineering beitragen [Rah16] und somit auch den Weg zurück aus den Systemen des Betriebs in die Planungssystemen ebnen. Ebenso sieht Spath et. al. diese Datenrückführung als „wesentliche[n] Stellhebel zur Verkürzung der Produktentstehungszeit und zur besseren Abstimmung der Kapazitätsnutzung“

[Spa+13].

Im Folgenden wird gezeigt, wie sich die mathematischen Konzepte von Graphen auf Linked Data abbilden lassen, welche vorhandenen Werkzeuge zur Graphtransformation es gibt und warum es dennoch sinnvoll ist, für Linked Data eine spezielle Graph-Transformationsengine zu erstellen.

### 5.7.1 Graph-Transformation

Graphen sind eine spezielle Art von Modellen, die auf einem Netzwerk von Knoten und Kanten aufbauen. Diese Art der Modelle lassen sich gut mit Hilfe von Graph-Transformationen bearbeiten und die Transformationsregeln deklarativ beschreiben.

#### 5.7.1.1 Abbildung mathematischer Graphen auf Linked Data

Die Konzepte der Graphentheorie können ohne Problem auf die Eigenschaften von Linked Data abgebildet werden, da in beiden Welten Graphen als Grundelement eingesetzt werden.

Eine RDF Node entspricht einem Knoten in mathematischen Graphen. Dieser ist benannt (typed). Für RDF-Literale hat er den Ausgangsgrad 0 und ist repräsentiert durch einen von vordefinierten und erweiterbaren Datentypen. Dazu gehört z.B. *@xsd:String*, *@xsd:Date* und *@xsd:Integer*. Für URI-Node bekommt der Knoten eine URI als Identifikator und besitzt einen Grad größer als 0, da jeder Knoten in mindestens eine Aussage einbezogen sein muss.

Eine RDF Property wird in mathematischen Graphen durch eine Kante ausgedrückt, die gerichtet ist und ebenfalls einen Identifikator besitzt. Named Graphs in Linked Data entsprechen Teilgraphen, die ebenfalls benannt sind. Somit können alle wichtigen Konzepte aus beiden Domänen ineinander überführt werden. Eine solche Überführung ist auch in Abbildung B.2 im Anhang zu sehen, in der in der oberen Hälfte ein Linked Data Modell mit zwei verschiedenen Namedgraphs (<http://demo.com/c1> und <http://demo.com/chem>) auf zwei Teilgraphen (*G1* und *G2*) abgebildet wird.

#### 5.7.1.2 Werkzeuge

Für die Transformation von Graphen gibt es eine Reihe von etablierten Werkzeugen, die unterschiedliche Fähigkeiten und Stärken haben.

GROOVE<sup>14</sup> ist ein graphischer Transformator, der auf einem objekt-

---

<sup>14</sup><http://groove.cs.utwente.nl/>

orientierten Ansatz mit Klassen und Instanzen aufbaut. Als interessantes Feature hat GROOVE die Möglichkeit der einfachen Explorierung aller möglichen Zustände.

AAG<sup>15</sup> ist ein Transformationswerkzeug, das auf attributierte Graphen ausgelegt ist. Der TGG-Interpreter<sup>16</sup> ist eine offene Software, die an der Universität Paderborn entwickelt wurde und mit der TGG umgesetzt werden können.

Mit Atom3<sup>17</sup> der Universität Madrid ist das Erstellen und Ändern von Graphen sowie das Anwenden unterschiedlichster Formalismen möglich. eMoflon<sup>18</sup> ist eine Sammlung von Java eCore-Klassen und Bibliotheken für die Erstellung von eigenen Anwendungen, die Graph-Transformationen benötigen. eMoflon unterstützt TGGs und bietet eine umfangreiche Dokumentation. Progress ist eine Sprache für die deklarative Spezifikation von Transformationen auf gerichteten Graphen, für die es eine Reihe von Engines und Cross-Compiler für verschiedene Programmiersprachen gibt [SWZ95].

Die meisten Graphtransformationswerkzeuge (AGG, TGG-Interpreter, Atom3, eMoflon, Progress) sind jedoch alle nur bedingt für Linked Data geeignet. Sie bauen auf typen-basierten Regeln auf. Hier wird eine Regel für ein Meta-Modell geschrieben, die im Modell nach Instanzen der jeweils spezifizierten Typen sucht und so die erforderlichen Matchings mit den dazugehörigen Verbindungen zwischen den Knoten herstellt. Dieses Vorgehen ist in den allermeisten Fällen sinnvoll und ermöglicht es, verständliche Regeln zu erstellen, ist jedoch in LD nicht unbedingt notwendig und immer zielführend. Auf Grund der Open-World-Assumption kann es Knoten geben, die keinen fest spezifizierten Typen besitzen, sondern vielmehr durch die Typen der benachbarten Knoten identifiziert werden. Dieses Problem tritt häufig in Zusammenhang mit Blank Nodes auf, denen man bewusst keinen eigenen Identifikatoren und Typ zuweist.

Einige Tools erlauben es zwar, auch untypisierte Knoten in Regeln zu verwenden, aber auch hier passt das Datenmodell mit dem Unterschied zwischen Literalen und Ressourcen nicht gut zu LD, weswegen im Folgenden ein Konzept für Graphtransformationen mit den Möglichkeiten und Technologien von Linked Data aufgezeigt werden soll.

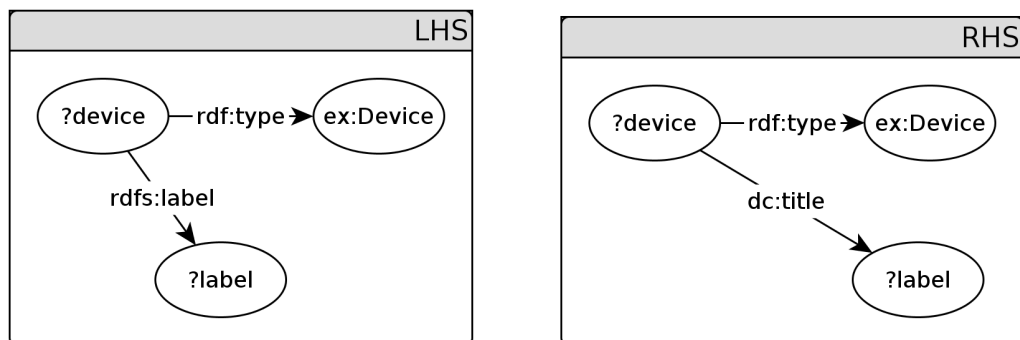
---

<sup>15</sup><http://user.cs.tu-berlin.de/~gragra/agg/>

<sup>16</sup><http://www.cs.uni-paderborn.de/index.php?id=tgg-interpreter>

<sup>17</sup><http://atom3.cs.mcgill.ca/>

<sup>18</sup><https://emoflon.github.io/>



**Abbildung 5.5:** Graphtransformationsregel zur Ersetzung des Prädikats *rdfs:label* mit *dc:title*

### 5.7.1.3 Graph-Transformation für Linked Data

Im Unterschied zu den meisten anderen formalen Graphersetzungssystemen baut die Graphtheorie von Linked Data jedoch nicht auf einem Knoten-basierten Ansatz auf, sondern ist komplett Kanten-zentriert. Dies bedeutet, dass Knoten nur durch die Referenzierung in einer Kante definiert sind. Daher kann es auch keine solche Unterscheidungen wie zwischen SPO (Single-Push-Out) und DPO (Double-Push-Out) geben. Es ist schlicht keine Möglichkeit vorhanden, einen Knoten zu löschen. Dies kann nur dadurch realisiert werden, dass alle Kanten mit Bezug zu diesem Knoten (als Subjekt oder Objekt) gelöscht werden, was natürlich auch einfach in SPARQL auszudrücken ist.

Das Konzept der Graph-Transformation lässt sich gut in das Konzept von LED integrieren [GU15]. Dabei kann SPARQL als Transformationsengine genutzt werden, da diese Sprache mit den entsprechenden INSERT und DELETE Queries im Zusammenspiel mit MINUS-Clauses über alle notwendigen Funktionalitäten für eine deklarative Transformationsbeschreibung und -ausführung verfügt. Somit kann jede Transformationsregel als einzelne – teilweise sehr komplexe – SPARQL-Abfrage formuliert werden.

Als Beispiel soll die Regel aus Abbildung 5.5 dienen. Diese sucht alle Geräte vom Typ *ex:Device* und ersetzt deren Property zur Benennung vom Typ *rdfs:label* durch den Typ *dc:title*<sup>19</sup>. Die ungebundene Variablen *?device* und *?label* verbinden hier LHS und RHS.

Diese graphisch spezifizierte operationale Transformationsregel kann mit Hilfe von SPARQL einfach ausgedrückt werden, wie es im Listing 5.3 zu sehen

<sup>19</sup>Die Präfixe *ex*, *dc*, *rdfs* und *rdf* wurden aus Platzgründen weggelassen.

**Listing 5.3:** SPARQL Repräsentierung der Transformationsregel aus Abbildung 5.5

---

```
INSERT {           # RHS-LHS
  ?device dc:title ?label.
}
DELETE {           # LHS-RHS
  ?device rdfs:label ?label.
}
WHERE {             # LHS
  ?device rdf:type ex:Device;
  rdfs:label ?label.
}
```

---

ist. Der **WHERE**-Abschnitt (Zeile 7–10) enthält die komplette LHS mit allen Tripeln, für die ein Matching für die Anwendung dieser Regel notwendig ist.

Der **INSERT**-Abschnitt (Zeilen 1–3) und der **DELETE**-Abschnitt (Zeilen 4–6) bilden zusammen mit dem **WHERE**-Abschnitt den RHS der Regel. Ersterer ist der Teil, der die Tripel enthält, die zur LHS hinzukommen und der zweite Teil enthält alle Tripel, die nur im LHS und nicht mehr im RHS auftauchen. Die Aufteilung erlaubt eine schnelle Unterscheidung in hinzufügende und löschende Regeln, so dass z.B. die Prüfung auf Monotonie der Graphtransformati-  
onsgrammatik simpel ist.

Negative Anwendungsbedingungen (NAC<sup>20</sup>) können durch entsprechende Filter-Operatoren (**FILTER NOT EXISTS ...**) in SPARQL ausgedrückt werden. Wertberechnungen und -zuweisungen in Transformationsregeln können durch **LET**-Konstrukte in SPARQL umgesetzt werden. Instanziierung einer Klasse (als typisierte Regeln) können durch einfache Kanten mit der URI *rdf:type* als Prädikat ausgedrückt werden. Dies ist aber nicht erforderlich, sodass Knoten auch anhand anderer Eigenschaften als dem Knotentyp ausgewählt werden können (z.B. über Beziehungen zu anderen Knoten).

### 5.7.2 Modell-Synchronisierung

Mit Hilfe der vorgestellten Modell-Transformationen ist es möglich, der Flexibilität in den Informationsmodellen und über den Verlauf der Zeit Herr zu werden. Ein natürliches Verhalten von LED-Datensätze besteht in der Anpassung und Erweiterung. Gleichzeitig ist die Verlinkung mit anderen Datensätzen essentiell. Somit müssen verschiedene Modelle miteinander syn-

---

<sup>20</sup>Negative Assertion Condition

chronisiert werden.

Hiermit einher geht auch die Fragestellung nach Schema-Evolution und Revisionierung. Wenn sich nun eine Ontologie weiterentwickelt und die Änderungen in einem Revisionverwaltungssystem adäquat vorgehalten werden, ist es möglich, die der Ontologie folgenden Instanzdaten an die Änderung anzupassen. Die Anpassung kann in einfachen Fällen vollautomatisch passieren. In anderen Fällen können aus den Änderungen in einer Ontologie aber zumindest Vorschläge für Transformationsregeln (in SPARQL) erstellt werden, die z.B. Umstrukturierungen von Properties abbilden können.

## 5.8 Prozesse für LED

### 5.8.1 Standardisierung vs. Flexibilität mit LED

Eine Standardisierung bei Informationsmodellen ist sinnvoll, denn dadurch kann sicher gestellt werden, dass verschiedene Organisationen kompatible Modelle erstellen und andere einfach nutzen können. Prinzipiell ist eine Standardisierung für das Verständnis der Daten nicht notwendig, wenn deren Bedeutung über eine semantische Beschreibung formal ausgedrückt werden kann. Auf der anderen Seite erleichtert ein gemeinsamer Standard die gemeinsame Nutzung der Konzepte, da keine (automatische) Interpretation und Transformation notwendig ist.

Eine wichtige Eigenschaft eines flexiblen und stabilen Informationsraumes ist dabei jedoch, dass es einen Ausgleich zwischen einer flexiblen Nutzung und der Standardisierung gibt. So benötigt man die Geschwindigkeit, um agil auf Anforderungen reagieren zu können. Desweiteren gibt es einen engen Zusammenhang zwischen Flexibilität von Informationsmodellen und der mit einem gewissen Aufwand erreichbaren Informationssicherheit für diese Systeme [Gra+13b]. In der realen Anwendung werden sich einige Informationsmodelle nach einigen Erweiterungen und Anpassungen durchsetzen. Diese sollten, wenn sie eine gewisse Stabilität erreicht haben, einer Standardisierung unterzogen werden. Dazu hat Drath (für die Nutzung in AutomationML) ein Vorgehensmodell mit verschiedenen Standardisierungsstufen erstellt [DB12]. Dies lässt sich ohne Probleme auf andere Informationsmodelle übertragen.

Da Linked Data nicht über ein Gremium standardisiert wird, sondern prinzipiell offen gestaltet ist, ergeben sich mehrere Möglichkeiten für die Festlegung von Ontologien. Zuerst ist dabei die W3C zu nennen, die bereits die Grundprinzipien und eine Menge der Basis-Ontologien definiert hat. Daneben ist es aber auch gut vorstellbar, die Standardisierung von



automatisierungstechnisch-relevanten Vokabularen in die domänenspezifischen Organisationen zu bringen. So könnte z. B. das internationale Elektronisches Vokabular (IEV) [IEC16] als kuriiertes Vokabular der Normen der IEC eine solche Rolle einnehmen.

Dazu sollte es möglich werden, die zu Grunde liegende Datenbank *Electropedia*<sup>21</sup> so zu erweitern, dass Begriffe über eine feste IRI abgerufen werden können. Dazu sollte das System um eine Repräsentation in RDF mit einem allgemeinen Vokabular erweitert werden. Entsprechende Vorschläge wurden im Zuge dieser Arbeit über das DKE an die IEC herangetragen.

Eine Alternative für die Bereitstellung von allgemein zugänglichen URIs für Objekte liegt z.B. in *eClass*<sup>22</sup>. Da diese bereits einen eindeutigen Identifikator sowohl für die Elemente als auch Attribute verfügen, ist die Veröffentlichung dieser Informationen als HTTP URI einfach zu erreichen. Dieser Schritt würde ein allgemein anerkanntes und kuriiertes Vokabular auf eine neue Qualitätsstufe heben, da es nun semantisch verwendet werden kann.

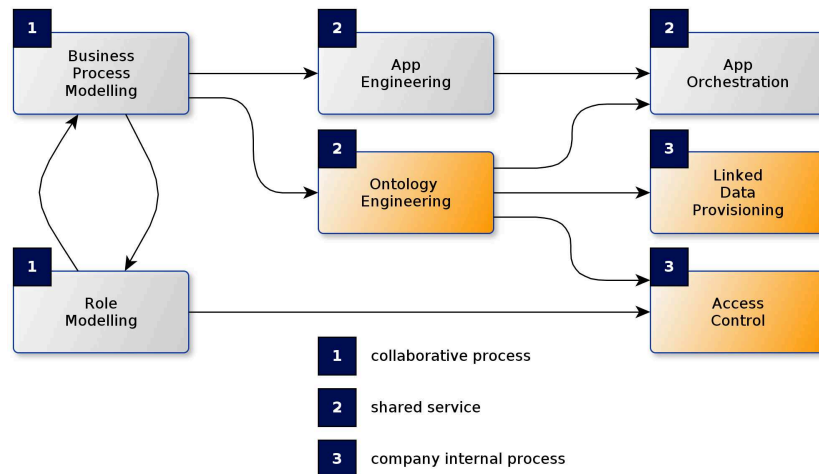
### 5.8.2 Einordnung von LED in die Geschäftsprozesse von Virtuellen Unternehmen

Zum Etablieren neuer Geschäftsmodelle in einem virtuellen Unternehmen kann Linked Enterprise Data die wichtige Rolle zur Bereitstellung des gemeinsamen Informationsraum erfüllen. Die Prozesse zum Integrieren verschiedener Unternehmen in ein virtuelles Unternehmen wird von Ziegler et. al. [Zie+14] ausführlich in einem siebenteiligen seriell parallelen Prozess beschrieben. Abbildung 5.6 bildet dabei die Teilprozesse, in denen die Linked Enterprise Data Cloud als gemeinsames Informationsmodells aufgebaut wird, in orange ab. In den Prozessen *Ontology Engineering*, *Linked Data Provisioning* und *Access Control* kann LED einen Beitrag zu einer effizienten und wartungsfreundlichen Gestaltung beitragen, wohingegen die anderen Prozesse entweder auf einer nicht-technischen Ebene anzutreffen sind (Business Process Modelling und Role Modeling) bzw. die Gestaltung des Frontends (App Engineering und App Orchestration) im Fokus haben. Bei Letzteren kann durch die Nutzung von LED ebenfalls eine effiziente Gestaltung unterstützt werden, was dabei aber nicht die entscheidende Rolle spielt.

---

<sup>21</sup><http://www.dke.r/de/Online-Service/DKE-IEV/Seiten/IEV-Woerterbuch.aspx>  
bzw. <http://www.electropedia.org/>

<sup>22</sup><http://www.eclass.eu/>



**Abbildung 5.6:** Linked Enterprise Data (orange) im Gesamtprozess eines virtuellen Unternehmens nach [Zie+14]

## 6 Implementierung - Notwendige Komponenten für LED

### Aufbau und Einordnung des Kapitels

Die in Kapitel 5 vorgestellten Konzepte für das eingeführte *Linked Enterprise Data* wurden zur Umsetzung eines integrierten Informationsraumes implementiert. Dieses Kapitel gibt eine Übersicht über die implementierten und genutzten Komponenten. Wesentlich ist dabei die Einrichtung einer geeigneten Architektur und die Einbindung vorhandener Daten aus bestehenden Systemen. Desweiteren werden Unterstützungswerkzeuge für Entwickler beschrieben, die LED-Applikationen entwickeln wollen. Das Kapitel wird mit der Beschreibung einiger Erweiterungen für die Laufzeit geschlossen, die Lösungen zu den Anforderungen der Revisionierung und Informationssicherheit angehen. Dabei bildet Tabelle 6.1 die im vorhergehenden Kapitel vorgestellt Konzepte auf einzelne Komponenten des Systems ab.

**Tabelle 6.1:** Abbildung der LED-Konzepte auf Laufzeitkomponenten

Konzept/Anforderung	Implementierung	Kapitel
Dezentrale Architektur (Kap. 5.3)	ComVantage-Architektur	6.1
Dynamische Adapter (Kap. 5.4.1)	OPCUA2LD	6.2
Statische Adapter (Kap. 5.4.2)	CESS	6.3
Informationssicherheit (Kap. 5.5)	SPARQL Security Rewriter	6.4
Versionierung (Kap. 5.6)	R43ples	6.5
Modell-Transformation (Kap. 5.7)	SMT	6.6
Inspizierbarkeit (Kap. 3.2.3.5)	RDF UML Diagramm	6.7

## 6.1 ComVantage-Architektur

Um die Anforderungen von virtuellen Unternehmen an ein Informationsraum auf Basis von Linked Enterprise Data zu erfüllen wurde im Projekt ComVantage eine verteilte Architektur erstellt, wie sie in Abbildung 6.1 zu sehen ist [Mün13; Gra+14a]. Dabei teilt sich die umgesetzte Architektur in ein Workflow-Layer, ein Integration-Layer und ein Data-Source-Layer auf, die durch definierte Schnittstellen miteinander verbunden sind und eine Trennung von verschiedenen Domänen und des Zugriffs auf Informationen erlauben.

In dem oben liegenden Workflow-Layer können einzelne spezialisierte Apps mit Hilfe von App-Orchestrierung so aneinander geschaltet werden, dass sie eine komplexe Gesamtaufgabe gut erfüllen können [Zie+12; Pfe+13; PU15]. Die Apps greifen dabei auf Informationen im Linked Data Format zurück und übergeben an folgende Apps den aktuellen Kontext per Linked Data URI.

Der Informationszugriff für die Apps wird durch den Integration-Layer gesteuert. In diesem sitzen mehrere Domain Access Server, die den Clients über ein organisationsübergreifendes Access Control Interface Zugriff auf die jeweils darunter liegenden Datenhaushalte bietet. Dabei werden die einzelnen Datenhaushalte über das Data Integration Layer zusammengeführt. Diese werden aus dem Data Source Layer über entsprechende Linked Data Adapter semantisch geliftet. Die Adapter können dabei auch bidirektional sein, also ein Schreiben in die Datenquelle aus dem Linked Data Haushalt ermöglichen. Dabei werden die beiden unteren Ebenen der ComVantage-Architektur durch Linked Enterprise Data erfüllt.

Eine beispielhafte Architektur einer Umsetzung wurde in dem ComVantage-Projekt umgesetzt und ist in Abbildung B.1 im Anhang dargestellt.

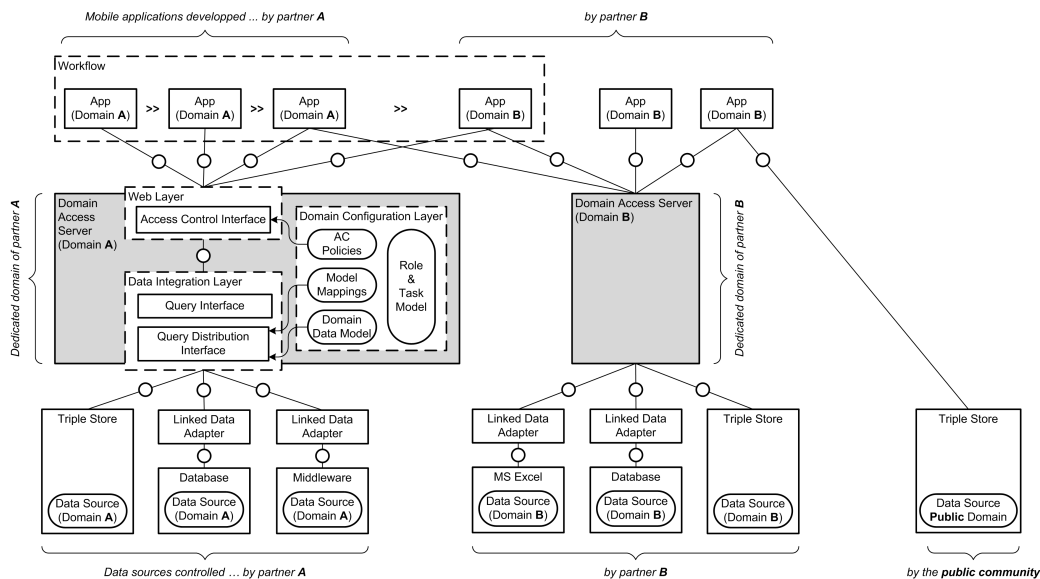


Abbildung 6.1: Überblick über die LED-Architektur

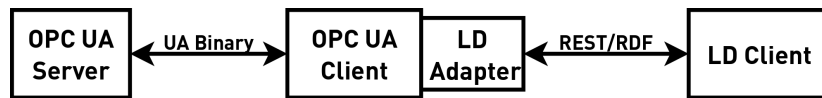
## 6.2 OPCUA2LD

Wie bereits im Analysekapitel aufgezeigt, ist OPC UA ein viel diskutierte Middleware-Lösung im Bereich von Industrie 4.0. Diese Stellung hat OPC UA auch zu Recht inne, da es viele Anforderungen an flexible Informationsräume gut abdecken kann. Es hat seine Stärken klar in der Abbildung und dem Austausch von dynamischen Daten in einer semantischen Struktur. Jedoch sind die statischen Abbildungen und die Verknüpfungen zu anderen Informationsträgern bei OPC UA nur sehr schwach ausgeprägt.

Auf Grund der ähnlichen Struktur der Basismodelle von Linked Data und OPC UA, die beide auf einen Graphen abgebildet werden können, ist es hier ein sinnvoller Ansatz, beide Welten zusammenzubringen.

Die hier vorgestellte Implementierung OPCUA2LD ist in einem iterativen Prozess mit Unterstützung von studentischen Arbeiten entstanden [Li12; Li13; Jun14; Gra+13a]. Die Studienarbeit von Li [Li12] beschäftigte sich mit der Frage, inwieweit Classic OPC als Vorgänger von OPC UA in ein Linked Data Netzwerk eingebunden werden kann. Dies wurde dann auf OPC UA ausgeweitet [Li13] und generalisiert [Jun14]

Das Konzept (siehe Abbildung 6.2) sieht einen Linked Data Client vor, der über REST Service mit dem Linked Data Adapter kommuniziert. Dieser agiert wiederum als OPC UA Client, der über das OPC UA Binary Protokoll



**Abbildung 6.2:** OPCUA2LD Adapter Konzept

mit dem OPC UA Server spricht und die REST-Anfragen in OPC UA Requests übersetzt. Das Ergebnis des OPC UA Requests wird durch den Linked Data Adapter im Body der HTTP-Response als RDF eingefügt. Dazu wird je nach Anfrage einer der gebräuchlichen RDF-Serialisierungen verwendet (z.B. RDF/XML, N3, Turtle, JSON-LD). Im Falle der Abfrage eines Wertes wird genau dieser Inhalt ausgegeben. Bei Methoden, die eine Änderung im Server bewirken, wird nur eine Rückmeldung über den Erfolg der Operation zurückgegeben.

### 6.2.1 Mapping zwischen OPC UA und Linked Data

In OPC UA gibt es mehrere Möglichkeiten, einen OPC UA Knoten zu identifizieren. Dies ist einerseits über die *NodeID* möglich, andererseits aber auch über den sogenannten *BrowsePath*, der die Einordnung des Knotens in die Hierarchie des Informationsraums beschreibt. Dabei wird hier jedoch die *NodeId* bevorzugt, da diese üblicherweise kürzer ist und desweiteren nahezu losgelöst von missleitender Semantik ist. Die *NodeId* besteht dabei aus einem Namespace, der für jeden Server durchnummeriert wird, sowie aus der spezifischen ID, die mit Hilfe verschiedener Typen ausgedrückt werden kann (z.B. `ns=2;i=84` für eine numerische *NodeId* aus dem Namespace 2 und `ns=3;s=Pump1` für eine String-Id aus dem Namespace 3).

Die Identifizierung einer Ressource erfolgt in Linked Data über HTTP URIs. Damit ist eine einfache direkte Abbildung zwischen beiden Welten möglich. Eine URI, die eindeutig einen OPC UA Knoten referenziert, kann z.B. folgendermaßen definiert werden:

`http://<server>/<path>/ns/<namespace>/<type>/<value>`

Ein alternatives Mapping kann mit URL Query Parametern dargestellt werden (`http://<server>/<path>/?ns=<namespace>&<type>=<value>`). Dieses entspricht doch weniger gut den REST-Prinzipien, die Query-Parameter eher als Parameter für Zugriffsmethoden als für die Spezifizierung einer Ressource sehen. Desweiteren ist die Path-Komponente einer URI für hierarchische Identifikatoren einer Ressource gedacht [BFM05], welche ein Namespace

und die NodeID sind, wohingegen der Query-Teil für eine nicht-hierarchische Identifikation gedacht ist.

Folgende Beispiele sollen den Ansatz weiter illustrieren: Die HTTP URI `http://eatld.et.tu-dresden.de/OPCUA2LD/ns/2/i/84` zeigt auf einen Knoten aus dem Server laufend auf der `http://eatld.et.tu-dresden.de` im Namespace 2 mit dem Integer-Identifizier `i=84`, wohingegen, `http://eatld.et.tu-dresden.de/OPCUA2LD/ns/3/s/PumpP001` auf eine Ressource im Namespace 3 und einem String-Identifizier mit dem Wert `PumpP001` zeigt.

### 6.2.2 Zugriffsmethoden

Diese Basisressource kann nun genutzt werden, um mit verschiedenen HTTP-Methoden und Query-Parametern auf die Services von OPC UA zuzugreifen. In den folgenden Beispielen wird *mw* als Präfix genutzt, der auf die Basis-URL des OPC UA Adapters zeigt. Mit diesem können die Mappings auf OPC UA Ressourcen einfach gestaltet werden.

#### 6.2.2.1 Browsing und Attribute lesen

Der Standardzugriff auf einen Knoten über einen GET-Request ohne zusätzliche Parameter gibt die Standard-Attribute sowie alle Referenzen des Knotens gemeinsam zurück. Dabei stellt Listing 6.2 das Ergebnis auf die Anfrage aus 6.1 dar. Hierbei ist wichtig, dass die Konzepte aus dem Namespace 0 von OPC UA, die fest standardisiert sind, auf feste URIs (z.B. *opcua:browseName* und *opcua:Object*) abgebildet werden, an denen die jeweiligen Konzepte in einer OWL-Ontologie beschrieben sind. Mit diesen können dann die Verweise auf die anderen Knoten abgebildet werden (hier z.B. *opcua:organizes*). Falls die Referenz nicht zum Namespace 0 gehört, wird stattdessen einfach die HTTP URI des Knoten verwendet (hier bei *mw:ns/2/s/isConnectedTo*).

---

**Listing 6.1:** GET Request für das Browsing eines Knotens

---

```
GET /OPCUA2LD/ns/2/s/TA1
Host: eatld.et.tu-dresden.de
```

---

---

**Listing 6.2:** Ergebnis für Anfrage aus Listing 6.1

---

```
HTTP/1.1 200 OK
Content-Type: text/turtle

mw:ns/2/s/TA1 a opcua:Object;
  opcua:browseName "Teilanlage 1";
```

```
opcua:hasTypeDefinition mw:ns/2/i/61;  
opcua:organizes mw:ns/2/i/25, mw:ns/2/s/pumpA;  
mw:ns/2/s/isConnectedTo mw:ns/2/s/TA2.
```

---

Wenn darüber hinaus der Knoten vom Typ Variable ist, wird das aktuelle Value-Attribut in spezieller Form zurückgegeben, um dieses semantisch gut auszudrücken zu können. Listing 6.3 ist eine Anfrage auf einen solchen Knoten und der Service hat als Ergebnis den Inhalt aus Listing 6.4.

Hier wird VOPR als Ontologie zur Darstellung der Messung verwendet. Dies ist ein selbst erstelltes Vokabular, das auf QUDT<sup>1</sup> und dem RDF Data Cube vocabulary<sup>2</sup> aufbaut und zusätzlich Verknüpfungen zwischen den REST-Ressourcen und dem Linked Data Service modellieren kann. So enthalten alle Ergebnisse wiederum einen Link auf den Service, der das Ergebnis erstellt hat (*vopr:providedBy*), sodass auch später klar ersichtlich ist, woher die Information kommt.

**Listing 6.3:** Ergebnis eines GET Requests zum Auslesen einer OPC UA Variable

---

```
GET http://eatld.et.tu-dresden.de/OPCUA2LD/ns/2/i/23  
Host: eatld.et.tu-dresden.de
```

---

**Listing 6.4:** Ergebnis für Anfrage aus Listing 6.3

---

```
HTTP/1.1 200 OK  
Content-Type: text/turtle  
  
mw:ns/2/i/23 a opcua:Variable;  
  opcua:browseName "Temperatur T001";  
  opcua:hasTypeDefinition mw:ns/2/i/62.  
  
[] a vopr:Value ;  
  vopr:value 23.0;  
  vopr:atTime "2016-04-12T10:50:28Z";  
  vopr:providedBy mw:ns/2/i/23.
```

---

### 6.2.2.2 Zugriff auf historische Daten

Ein Zugriff auf historische Daten erfolgt ebenfalls durch einen GET-Request auf die Basis-Ressource, der um die Query-Parameter *start* und *end* ergänzt wird. Diese schränken das Zeitfenster für die Werte einer Variablen ein. So zeigt der Aufruf von Listing 6.5 das Ergebnis aus Listing 6.6. Die Ergebnisse

---

<sup>1</sup><http://www.qudt.org>

<sup>2</sup><http://www.w3.org/TR/vocab-data-cube/>



werden wiederum in VOPR zurückgegeben, die die Slices aus dem RDF Data Cube Vocabulary verwenden, um die Serie an Messwerten zurückzugeben.

**Listing 6.5:** Ergebnis eines GET Requests für den Zugriff auf historische Daten

---

```
GET http://eatld.et.tu-dresden.de/OPCUA2LD/ns/2/i/23?start
    =2016-04-09T154045\&end=2016-09-04T154445
Host: eatld.et.tu-dresden.de
```

---

**Listing 6.6:** Ergebnis für Anfrage aus Listing 6.5

---

```
HTTP/1.1 200 OK
Content-Type: text/turtle

mw:ns/2/i/23 a opcua:Variable;
  opcua:browseName "Temperatur T001";
  opcua:hasTypeDefinition mw:ns/2/i/62.

[] a vopr:HistorySlice ;
  vopr:startAt "2016-04-09T15:40:45.177";
  vopr:endAt "2016-04-09T15:44:45.177" ;
  vopr:hasHistory[
    a rdf:Seq ;
    rdf:_1 [ a vopr:Value ;
      vopr:atTime "2016-04-09T15:43:50.375" ;
      vopr:value "67.0"] ;
    rdf:_2 [ a vopr:Value ;
      vopr:atTime "2016-04-09T15:43:49.375";
      vopr:value "66.0"];
  ];
  vopr:providedBy mw:ns/2/i/23.
```

---

### 6.2.2.3 Werte schreiben

Wenn Variablen in OPC UA schreibbar sind, können diese auch durch den Linked Data Adapter geändert werden. Dieses funktioniert über einen POST-Request mit dem gewünschten Wert als Parameter im Request-Body (siehe Listing 6.7). Als Ergebnis gibt es darauf nur einen booleschen Rückgabewert, ob das Schreiben erfolgreich war (siehe Listing 6.8).

**Listing 6.7:** POST Request zum Schreiben einer OPC UA Variable

---

```
POST /OPCUA2LD/ns/2/i/100
Host: eatld.et.tu-dresden.de

value=23
```

---

**Listing 6.8:** Ergebnis zu Listing 6.7

---

```
HTTP/1.1 200 OK
Content-Type: text/boolean

true
```

---

#### 6.2.2.4 Methoden aufrufen

Falls der entsprechende OPC UA Knoten existiert und eine Methode darstellt, also vom Typ *Method* ist, führt ein POST-Request zum Aufruf derselben. Die Parameter für die Methode werden im Body mitgesendet (siehe Listings 6.9 und 6.10).

**Listing 6.9:** POST Requests für den Aufruf einer OPC UA Methode

---

```
POST /OPCUA2LD/ns/2/i/100
Host: eatld.et.tu-dresden.de

a=3&b=6
```

---

Die Rückgabewerte der Methode werden in den Body der HTTP-Response übertragen und stellen sich z.B. für eine Additionsmethode wie in Listing 6.10 dar.

**Listing 6.10:** Ergebnis für den Aufruf der Methode aus Listing 6.9

---

```
HTTP/1.1 200 OK
Content-Type: text/plain

c=9
```

---

#### 6.2.2.5 Nicht unterstützte OPC UA Dienste

Die OPC UA Service Sets für Discovery, SecureChannel und Session werden automatisch im Hintergrund des Adapters behandelt und können durch den Nutzer nicht explizit aufgerufen werden. Das bedeutet also, dass sich der Nutzer nicht um die Verbindungserstellung kümmern muss.

Die Service Sets für *MonitoredItem* und *Subscription* können nicht unterstützt werden, da diese gegen die Forderung der Zustandslosigkeit in REST-Services verstoßen.

Alle Dienste, die die Struktur vom OPC UA Namensraum ändern, also das Hinzufügen und Löschen von Knoten und Referenzen, können prinzipiell durch HTTP PUT bzw. HTTP DELETE Anfragen abgebildet werden.

Davon wird in diesem Konzept jedoch explizit aus Gründen der Informationssicherheit Abstand genommen, auch wenn eine HTTP Verbindung mit den entsprechenden Authorisierungsmechanismen abgesichert werden kann. Hierbei entstünde aber eine sehr starke Verkopplung der Informationsräume, deren Auswirkungen nicht mehr einfach zu überschauen sind.

Der Dienst *HistoryUpdate* aus dem Attribute Service Set ist ebenfalls nichtunterstützt genauso wie *BrowseNext*, *TranslateBrowsePathsToNodeIds*, *RegisterNodes* und *UnregisterNodes* aus dem View Service Set. Das Query Service Set wurde noch nicht implementiert. Hier bietet sich jedoch die Möglichkeit an, dieses auf SPARQL abzubilden und somit einfacher komplexe Queries zu ermöglichen, als es in OPC UA derzeit möglich ist.

### 6.2.3 Implementierung

Das vorgestellte Konzept wurde in einer prototypischen Java-Applikation<sup>3</sup> umgesetzt. Diese verwendet für den Zugriff auf OPC UA das Prosys OPC UA SDK<sup>4</sup> und Jersey<sup>5</sup> als Framework für die Umsetzung des REST-Interface.

## 6.3 CESS - COMOS Enterprise Server Service

CESS (COMOS Enterprise Server Service) ist eine beispielhafte prototypische Implementierung für einen quasi-statischen Linked Data Adapter. CESS kann dabei Informationen aus dem Anlagenplanungswerkzeug COMOS<sup>6</sup> auslesen und als Linked Data auf einem Triplestore bereitstellen.

CESS ermöglicht den Zugriff auf Anlagendaten, welche im CAE-Tool COMOS vorgehalten werden. Der Service dient als zentraler Zugriffspunkt für alle Arten von Clients, die mittels HTTP auf diese Daten zugreifen möchten. Durch die unterstützten Import- und Exportfunktionalitäten wird der bidirektionale Datenaustausch zwischen Client und COMOS gewährleistet, welcher sich jedoch auf die in COMOS definierten Konnektoren beschränkt. Als Austauschformat wird vorrangig XML genutzt, wobei der CESS auch die Bereitstellung und Verarbeitung von Daten in der RDF-Serialisierung Turtle unterstützt, die sich jedoch auf ausgewählte Konnektoren beschränkt. Der

---

<sup>3</sup>[svn://dev.plt.et.tu-dresden.de/srv/svn/repositories/LinkedData/trunk/LinkedData/OPC2LD/OPCUA2LD/trunk/OPCUA2LDService](https://svn://dev.plt.et.tu-dresden.de/srv/svn/repositories/LinkedData/trunk/LinkedData/OPC2LD/OPCUA2LD/trunk/OPCUA2LDService)

<sup>4</sup><https://www.prosysopc.com/products/opc-ua-java-sdk/>

<sup>5</sup><https://jersey.java.net/>

<sup>6</sup><http://comos.de>

CESS ist die Weiterentwicklung des *ComosMotionXServices* aus [Gra09] und die gleichzeitige Integration des daraufhin entwickelten *LinkedDataExporters*.

CESS nutzt für den bidirektionalen Datenaustausch mit COMOS den von COMOS mitgelieferten COMOS Enterprise Server, welcher durch sogenannte Kommandodateien Import- und Exportvorgänge auf Basis der in COMOS definierten Konnektoren ausführen kann (siehe Verteilungsdiagramm in Abbildung B.4 im Anhang). Der COMOS Enterprise Server benötigt vom COMOS License Server eine Lizenz, welche durch den COMOS License Manager bereitgestellt wird. Der Datenaustausch erfolgt durch den Zugriff auf den Datenbankserver. Clients besitzen die Möglichkeit, auf der Basis von HTTP auf die Funktionalitäten des CESS zuzugreifen.

Neben dem reinen XML-Export kann CESS auch eine anschließende Transformation der XML-Datei in eine RDF-Serialisierung mit einem Python-Skript vornehmen. Dieser Ablauf ist im Sequenzdiagramm in Abbildung 6.3 dargestellt. Einige der Konnektoren exportieren zusätzlich Dokumente, die als ZIP-Archiv über die REST-Schnittstelle heruntergeladen werden können. CESS verwendet dabei *Grizzly*<sup>7</sup> als zugrundeliegenden HTTP-Server, *Jersey*<sup>8</sup> zur Erstellung des RESTful Web Services und *Gson*<sup>9</sup> für die JSON-Serialisierung von Java Objekten.

---

<sup>7</sup><https://grizzly.java.net/>

<sup>8</sup><https://jersey.java.net/>

<sup>9</sup><https://code.google.com/p/google-gson/>

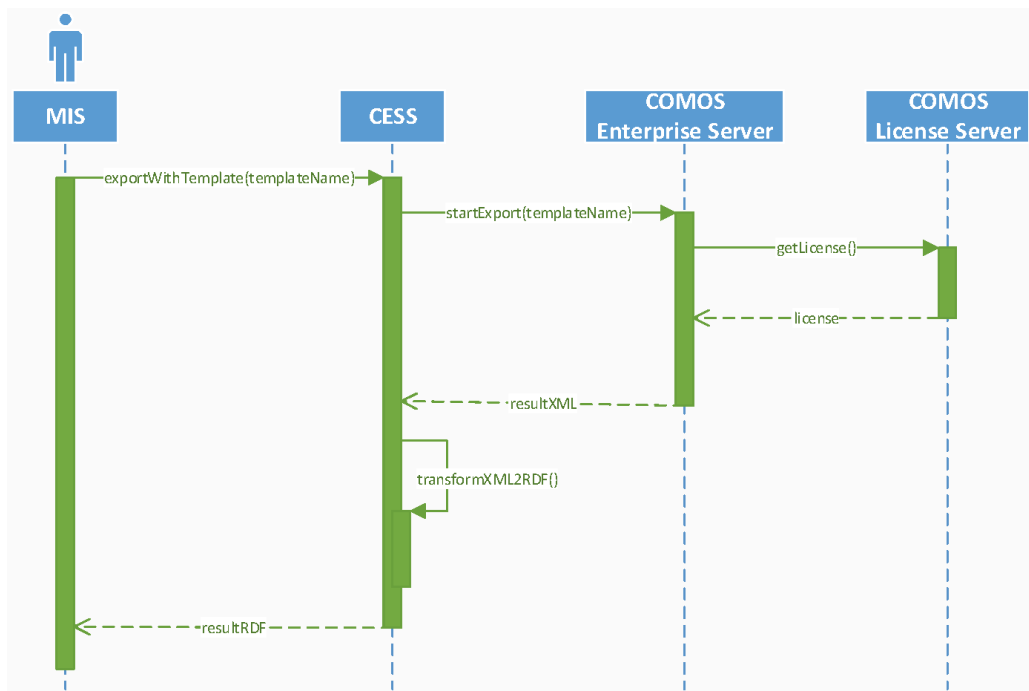


Abbildung 6.3: CESS Sequenzdiagramm für den Export von RDF

## 6.4 SPARQL Security Rewriter

Der Ansatz zur Informationsabsicherung in Linked Data wurde initial im ComVantage-Projekt [Ort+13] konzipiert und prototypisch umgesetzt. Dieses Konzept wird dann in [Gra+13b; Gra+13b; Gra+14b] auf die Vereinbarkeit mit flexiblen Informationsmodellen untersucht. Dabei kann gezeigt werden, dass eine geschickte Erstellung von Zugriffsregeln die Weiterentwicklung des Informationsmodells nicht hindert und die Zugriffsregeln ganz ohne oder nur mit geringen Aufwand auf ein neues Informationsmodell aktualisiert werden können.

In dem Ansatz von [Ort+13] wird zur Darstellung der Views im Triplestore und daher auch beim Query-Rewriting eine Reifikation genutzt nach [HP14] genutzt. Dabei wird jedes Tripel durch vier andere Tripel ausgedrückt mit *ac:subject*, *ac:predicate* und *ac:object* als Subproperties der entsprechenden RDFS-Konzepte<sup>10</sup>, um einem Tripel einen Kontext über eine URI geben zu können, auf dem die Zugriffsrechte festgelegt werden können. Dies funktioniert

<sup>10</sup><https://www.w3.org/TR/rdf11-mt/#reification>

**Listing 6.11:** Query zur Abfrage der Bekannten von *ins:jsmith* mit Namen und optionaler Mail-Adresse

---

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ins: <http://www.comvantage.eu/instances#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT DISTINCT ?name ?mail
WHERE {
    ins:jsmith foaf:knows ?y.
    ?y rdfs:label ?name.
    OPTIONAL {?y foaf:mbox ?mail}.
}
```

---

aber einfacher und ressourcensparender, wenn dies über Named Graphs gelöst wird, die jedem Tripel bereits einen Kontext mit übergeben. Daher wird im Folgenden der Schritt von Reifikation zu Named Graphs verfolgt.

Übergeordnet steht dabei ein auf XACML-basierendes organisationsübergreifendes Authentisierungsverfahren, das es Nutzern erlaubt, sich bei ihrem lokalen Identity-Provider anzumelden und mit dem erhaltenen Identity-Token bei den Service-Providern der Partner anmelden zu können. Dabei kann der Zugriff dynamisch durch weitere Attribute angepasst werden (z.B. Tageszeit, Anlagenzustand). Der Zugriff auf Informationen erfolgt dabei sowohl in der eigenen Organisation als auch bei den Partnern über einen abgesicherten SPARQL-Endpunkt. Der Client stellt eine SPARQL-Abfrage und der Service-Provider kann aus dem Identity-Token eine URI herauslesen, die den Nutzer eindeutig identifiziert, und daraufhin die SPARQL-Query so umschreiben, dass diese nur noch Zugriff auf für diesen Nutzer freigegebene Views enthält. Weiterhin wird der Schritt von der N3-Logik zur Generierung von Views zu SPARQL getätigt.

Beispielhaft wird die Abfrage in Listing 6.11 betrachtet, die alle Kontakte *?y* mit ihren Namen *?name* sowie der Mail-Adresse *?mail*, wenn vorhanden abfragt. Die Abfrage wird dabei auf einen kleinen Datensatz mit drei Personen gestellt, von denen *ins:pshaw* keine Mail-Adresse besitzt (siehe Listing B.2 im Anhang). Wird nun die Abfrage ohne vorherige Informationsabsicherung durchgeführt, führt zu dem Ergebnis aus Listing 6.2.

Nun sollen jedoch sowohl Mail-Adressen und Geburtstage als sensitive Daten nur von privilegierten Nutzern einsehbar sein. Daher wird zunächst der Rolle *role:operator* der Zugriff auf die anderen Properties gewährt, in dem

**Tabelle 6.2:** Ergebnis zur Abfrage aus Listing 6.11 ohne Informationsabsicherung

name	mail
Arthur Harris	aharris@dresscode21.com
Peter Shaw	

**Tabelle 6.3:** Ergebnis zur Abfrage aus Listing 6.15 mit einer Einschränkung auf die sichtbaren Views von *user:operator23*

name	mail
Arthur Harris	
Peter Shaw	

der View *view:operator1* durch die SPARQL-Query in Listing 6.12 erzeugt wird.

Für *role:manager* sollen jedoch auch diese Daten zur Verfügung stehen. Dies wird durch Listing 6.13 erledigt. Zudem wird noch die Zuordnung der Nutzer *user:operator23* und *user:manager12* zu den entsprechenden Rollen in Listing 6.14 festgelegt und im Graphen *cv:access-policies* gespeichert.

Der abfragende Nutzer soll in diesem Fall *role:operator23* sein. Daher wird die Original-Query für diesen Nutzer umgeschrieben, wobei in diesem Schritt noch nicht auf die Instanzen aus dem Triplestore zurückgegriffen werden muss. Das Ergebnis befindet sich in Listing 6.15. Dabei werden in den Zeilen 9–17 alle zugreifbaren Views abgefragt und dann in den Zeilen 18–25 jeweils geprüft, ob sich jedes Tripelstatement in einem dieser Views befindet. Dabei unterscheidet sich das Ergebnis für verschiedene Nutzer nur durch den Austausch von *user:operator23* durch die entsprechende URI eines anderen Nutzers und ist unabhängig von dem dahinterliegenden Datensatz.

Wenn die umgeschriebene Query nun ausgeführt wird, zeigt das Ergebnis aus Listing 6.3 tatsächlich, dass auf die Mail-Adresse nicht mehr zugegriffen werden kann. Die Ausführung der für *user:manager12* umgeschriebenen Query bringt aber dasselbe Ergebnis wie Listing 6.2.

Das Konzept wurde in Java unter Nutzung von Jena implementiert und ist zu finden unter <http://gitlab.plt.et.tu-dresden.de/mgraube/SparqlRewrite>. Dort befindet sich ebenfalls das vorgestellte Beispiel sowie weitere Demo-Applikationen.

**Listing 6.12:** Generierung des Views *view:operator1*

---

```
PREFIX  rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX  ac:   <http://www.comvantage.eu/ac-schema#>
PREFIX  role: <http://www.comvantage.eu/groups#>
PREFIX  view: <http://www.comvantage.eu/view/>
PREFIX  cv:   <http://www.comvantage.eu/>

INSERT {
  GRAPH cv:access-policies {
    role:operator ac:canSee view:operator1.
  }
  GRAPH view:operator1 {
    ?p a foaf:Person;
    rdfs:label ?name;
    foaf:firstName ?firstName;
    foaf:lastName ?lastName;
    foaf:knows ?p2.
  }
}
WHERE {
  ?p a foaf:Person;
  rdfs:label ?name;
  foaf:firstName ?firstName;
  foaf:lastName ?lastName;
  foaf:knows ?p2.
}
```

---



**Listing 6.13:** Generierung des Views *view:manager*

---

```
PREFIX  rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX  ac:   <http://www.comvantage.eu/ac-schema#>
PREFIX  role: <http://www.comvantage.eu/groups#>
PREFIX  view: <http://www.comvantage.eu/view/>
PREFIX  cv:   <http://www.comvantage.eu/>

INSERT {
  GRAPH cv:access-policies {
    role:manager ac:canSee view:manager.
  }
  GRAPH view:manager {
    ?p foaf:mbox ?mbox;
      foaf:birthday ?birthday.
  }
}
WHERE {
  ?p a foaf:Person;
    foaf:mbox ?mbox;
    foaf:birthday ?birthday.
}
```

---

**Listing 6.14:** Zuweisung von Nutzern zu Views

---

```
PREFIX  rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX  role: <http://www.comvantage.eu/groups#>
PREFIX  cv:   <http://www.comvantage.eu/>
PREFIX  user: <http://www.comvantage.eu/user#>

INSERT DATA {
  GRAPH cv:access-policies {
    user:operator23 rdfs:member role:operator.
    user:manager12  rdfs:member role:manager.
    user:manager12  rdfs:member role:operator.
  }
}
```

---

**Listing 6.15:** Umgeschriebene Query aus Listing 6.11 für den Nutzer *user:operator23*

---

```
1 PREFIX ac: <http://www.comvantage.eu/ac-schema#>
2 PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
3 PREFIX role: <http://www.comvantage.eu/ac-schema/groups#>
4 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
5 PREFIX ins: <http://www.comvantage.eu/instances#>
6
7 SELECT DISTINCT ?name ?mail
8 WHERE {
9     GRAPH <http://www.comvantage.eu/access-policies>
10     { user:operator23 rdfs:member ?group1 .
11       ?group1 ac:canSee ?view1 .
12       user:operator23 rdfs:member ?group2 .
13       ?group2 ac:canSee ?view2 .
14       user:operator23 rdfs:member ?group3 .
15       ?group3 ac:canSee ?view3 .}
16     GRAPH ?view1
17     { ins:jsmith foaf:knows ?y .}
18     GRAPH ?view2
19     { ?y rdfs:label ?name .}
20     OPTIONAL
21     { GRAPH ?view3
22       { ?y foaf:mbox ?mail .}
23     }
24 }
```

---

## 6.5 R43ples

R43ples<sup>11</sup> (Revision for Triples) ist eine Implementierung zur Versionsverwaltung im Semantic Web, das auf einer konsequenten Nutzung einer semantischen Beschreibung der Revisionen und der Unterschiede zwischen Revisionen beruht. Dabei baut es auf den Vorarbeiten von Vander Sande et. al. [Van+13] auf und erweitert diese zu einem komplett semantischen Informationsmodell, so dass kein implizites Wissen zur Herstellung alter Revisionen notwendig ist. R43ples ist aus den Vorarbeiten einer Studienarbeit [Hen13] entstanden, in der Synchronisierungsmechanismen zwischen verschiedenen bidirektionalen Linked Data Adaptern untersucht wurden. Die Dokumentation von R43ples steht auf der Webseite unter <http://plt-tud.github.io/r43ples/> bereit und der Quellcode ist unter EUPL-Lizenz auf GitHub unter <https://github.com/plt-tud/r43ples> verfügbar.

### 6.5.1 Konzept und Architektur

R43ples stellt einen SPARQL Proxy bereit [GHU14; GHU16]. Dieser nimmt sowohl normale SPARQL-Anfragen an, als auch um Revisionsinformationen erweiterte SPARQL-Abfragen, sogenannte R43ples-Queries. Wie in Abbildung 6.4 verdeutlicht werden die Abfragen für die Revisionsverwaltungsaufgaben umgeschrieben und an den SPARQL-Endpoint eines angeschlossenen Triplestores gesendet. Dieser übernimmt die komplette Datenhaltung und das Verwalten der semantischen Informationen. R43ples selbst hingegen hat keinerlei internen Speicher und ist zustandslos aufgebaut.

Die Datenstruktur von R43ples ist komplett semantisch aufgebaut und wird in dem angehängten Triplestore vorgehalten. Dabei definiert die Revision Management Ontology (RMO) Revisionen (*rmo:Revision*) und die dazu gehö-

<sup>11</sup>Sprechweise: /ˈrɪpəls/



**Abbildung 6.4:** Architektur von R43ples als SPARQL-Proxy

rigen Commits (*rmo:Commits*) basierend auf PROV [W3C13] als *prov:Entity* und *prov:Activity* abgebildet (siehe Abbildung B.7 im Anhang). An die entsprechenden Ressourcen werden die notwendigen Meta-Informationen und Verknüpfungen zu den Changesets in weiteren Named Graphs angehängt. Desweiteren können damit auch *rmo:Tags* und *rmo:Branches* abgebildet werden.

### 6.5.2 R43ples Interface

Der Nutzer kann mit R43ples über erweiterte SPARQL-Abfragen interagieren. Damit ist es möglich, einerseits neue Revisionen mit einem **SPARQL INSERT** oder **SPARQL DELETE** anzulegen und neue Branches und Tags einzufügen. Andererseits kann man auch alte Revisionen und Tags wieder abrufen. Dabei stellt R43ples die alten Informationen transparent für den Nutzer wieder her. Dafür wurden zwei verschiedene Ansätze implementiert. Der erste Ansatz geht von dem aktuellen Branch aus, für den immer eine vollständige Kopie des kompletten Graphen vorgehalten wird, kopiert diesen in einen temporären Graphen und wendet dann rückwärts alle Changesets auf die Kopie an bis die gewünschte Revision erreicht ist. Nun kann die Abfrage des Nutzers auf der Kopie ausgeführt und das Ergebnis zurückgeliefert werden. Dieser Ansatz ist relativ einfach und robust, benötigt aber bei großen Graphen für die Kopie viel Zeit. Daher wurde ein zweiter Ansatz implementiert, der durch ein Umschreiben der SPARQL-Query diese dann direkt auf den Triplestore ausführen kann und die richtigen Ergebnisse auch für alte Revisionen erhält. Dies behandelt, ähnlich wie bei dem SPARQL Security Rewriting, jedes Triplestatement separat und fügt für dieses abhängig von der gewählten Revision Joins von dem kompletten Graphen und den notwendigen Changesets ein. Details zu dem Konzept und der Implementierung vom Query Rewriting von R43ples finden sich in [GHU16].

Als graphische Oberfläche wird eine HTML5-Applikation verwendet (siehe Abbildung B.8 im Anhang). Diese ermöglicht den Zugriff auf alle Funktionen des Systems. Insbesondere ist es damit möglich, R43ples-Abfragen zu formulieren und auszuwerten, Revisionsgraphen anzuzeigen, Testdatensätze zu erzeugen und direkte Abfragen auf dem angehängten Triplestore auszuführen. Eine Oberfläche zur Nutzung der Merging-Funktionalität von R43ples ist in Arbeit [Hen14; Yan15].

### 6.5.3 Merging-Funktionalitäten

In weiteren Arbeiten [Hen14; Yan15; HGU16] wurde R43ples um ein System zur Konflikterkennung und -behebung erweitert. Dieses kann nach weiteren Ergänzungen nun Merging in den Modi *Three-Way-Merge*, *Rebase* und *Fast-Forward* durchführen. Da hierbei Konflikte auftreten, also Änderungen in den beiden zusammenzuführenden Branches, die sich gegenseitig widersprechen, wurde dafür eine Anwendung konzipiert und prototypisch umgesetzt, die dem Nutzer eine Visualisierung und Entscheidungsunterstützung anbietet, welches Triples aus welchem Branch er wählen möchte.

### 6.5.4 Leistungsmetriken

Wichtige Metriken für die Bewertung von Revisionsverwaltungssystemen sind die Speicheraufwände und die Antwortzeiten bei der Abfrage von alten Revisionen [GHU14].

#### 6.5.4.1 Speicherbedarf

Die Menge  $S_{\text{Revision}}$  von zusätzlichen Tripeln, um eine neue Revision  $i$  in dem System zu speichern, ist nahezu proportional zu der Größe der Änderungen ( $S_{\text{Changes}}^i = S_{\text{Add}}^i + S_{\text{Del}}^i$ ). Sie ist dabei komplett unabhängig von der Komplexität vorhergehender Revisionen oder dem Verlauf des Revisionsgraphen (6.1). Die konstante Anzahl zusätzlicher Tripel ist üblicherweise gegenüber der Größe des Changesets vernachlässigbar. Die Erstellung eines Branch (6.2) oder das Tagging einer Revision  $i$  (6.3) ist relativ teuer, da es eine komplette Kopie des vollständigen Graphen ( $S_{\text{Graph}}^i$ ) speichert, um einen schnellen Zugriff zu ermöglichen. Darüber hinaus wird noch eine geringe Anzahl an festen Tripeln gespeichert.

$$S_{\text{Revision}}^i = S_{\text{Changes}}^i + 12 \quad (6.1)$$

$$S_{\text{Branch}}^i = S_{\text{Graph}}^i + 15 \quad (6.2)$$

$$S_{\text{Tag}}^i = S_{\text{Graph}}^i + 11 \quad (6.3)$$

#### 6.5.4.2 Antwortzeiten für die Wiederherstellung alter Revisionen

Eine wichtige Einflussgröße auf die Gebrauchstauglichkeit eines Revisionierungssystems und des dahinter liegenden Konzepts ist die Antwortzeit des

Systems. Daher wurde die Zeit vom Start der Anfrage bis zum Empfang der Ergebnisse in verschiedenen Konfigurationen gemessen.

Um die Skalierungseffekte des Systems untersuchen zu können, wurden verschiedene künstliche Datensätze erstellt. Dazu wurden zufällige Datensätze mit jeweils 100, 1k, 10k, 100k und 1M Tripeln generiert. Für diese wurden ebenfalls zufällige Changesets für 21 Revisionen in den Größen von 10 bis 100 Tripeln erzeugt und in allen Varianten in R43ples als Revisionsgraphen geladen. Somit ergeben sich Szenarien mit den unterschiedlichen Dimensionen *Graphengröße*, *Größe des Changesets*, *abgefragte Revision* sowie *Art der Revisionserstellung* (Query Rewriting oder Erstellung eines temporären Graphen).

Eine einfache SPARQL-Abfrage (Suchen von allen Attributen für eine bestimmte Ressource) wurde zwanzig Mal durchgeführt, um zufällige Effekte wie Last durch weitere Prozesse und Garbage Collection zu kompensieren. Die Bearbeitungszeit der Anfragen an R43ples wurde in einem Setup mit einem 4 GB RAM System getestet, auf dem Stardog 4.0<sup>12</sup> als SPARQL Endpoint läuft.

Abbildung B.9 zeigt, dass bei dem Ansatz mit der Erzeugung eines temporären Graphen die Größe des Datensatzes sowie die Menge der zurückgegangenen Revisionen den größten Einflussfaktor darstellen [GHU14]. Während Revision 0 direkt auf dem Graphen ausgeführt werden kann, muss für jede andere Revision in der Vergangenheit zuerst der gesamte Datensatz kopiert werden, was zu dem Anstieg führt. Dieser Anstieg hängt stark von der Gesamtgröße des Datensatzes ab, wie die rechte Teilabbildung zeigt.

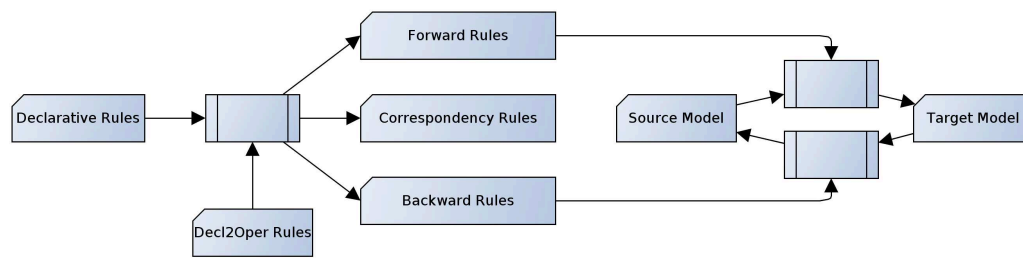
Auf der anderen Seite ist die Query Rewriting Option weniger abhängig von diesen Faktoren, sondern stärker von der Menge der Revisionen, die man in die Vergangenheit gehen möchte. Zudem gibt es hier deutliche Unterschiede in der Leistungsfähigkeit in Abhängigkeit von dem verwendeten Triplestore. Und zu guter Letzt hängt die Geschwindigkeit sehr stark von der Art der Abfrage ab. So wird eine Abfrage nach allen Tripeln in dem Datensatz mit der Query Rewriting Option länger dauern als die Erstellung des temporären Graphen, auf dem diese Abfrage dann wiederum sehr schnell ist.

### 6.6 SMT (SPARQL Model Transformation)

SMT (SPARQL Modell Transformation) ist eine entwickelte Methodik und Werkzeugkette, um direkt in Linked Enterprise Data semantische Modell-

---

<sup>12</sup><http://stardog.com/>



**Abbildung 6.5:** Konzept für SPARQL Transformation

transformationen durchzuführen, die auf den Konzepten aus Kapitel 5.7 aufbaut.

Abbildung 6.5 zeigt das Konzept für den SPARQL Modell Transformationsansatz, der auf Triple Graph Grammars (TGG) aufbaut. Dabei wird SPARQL sowohl für die Anwendung der operationalen TGG-Regeln genutzt, als auch für die Überführung von den deklarativen TGG-Regeln in operationale Regeln. Für diesen Schritt ist eine Umwandlung der SPARQL-Regeln in die SPIN (SPARQL Inference Notation) Syntax vorgenommen worden. Diese bildet eine textuelle SPARQL-Query in einem RDF-Modell ab. Für die Umwandlung stehen APIs für Java und Online-Konverter bereit. Das RDF-Modell der Query ist dann durch die Anwendung von 10 SPARQL-Transformationsregeln in ein Modell für eine der möglichen operationalen Regeln umgewandelt worden (links nach rechts, rechts nach links, Prüfung der Korrespondenz). Letztendlich wird das RDF-Modell über SPIN wieder in eine textuelle SPARQL-Abfrage überführt. In dieser Form wird es über eine normale SPARQL-Engine auf die eigentlichen Modelle angewendet.

Der Transformationsansatz mit SPARQL kann als Web Service implementiert werden, was durch eine geschickte REST-Schnittstelle eine ideale Integration in das LD-Universum erlaubt. Für die Nachweiseibringung der Nutzbarkeit eines solches Systems wurde hier jedoch vereinfacht eine lokale Anwendung implementiert. Der Quellcode des Ansatzes steht bei GitHub<sup>13</sup> zur freien Verfügung. Dazu wurde Apache Jena<sup>14</sup> Framework erfolgreich als Transformationsengine genutzt. Die RDF-Modelle (für die Source- und Target-Modelle und auch für die Überführung der deklarativen Regeln auf operationale Regeln) wurden dazu in einer TDB-Datenbank vorgehalten.

Die Implementierung wurde erfolgreich auf das Beispiel von [Kön05] angewendet. In diesem werden UML-Klassendiagramme in SQL Schemata und

<sup>13</sup><https://github.com/plt-tud/sparql-model-transformation>

<sup>14</sup><http://jena.apache.org/>

zurück transformiert. Ebenso wurden die akademischen Beispiele zu RailCab-Petrinetz durchgeführt [KW07]. Damit kann die generelle Anwendbarkeit des Ansatzes gezeigt werden.

Der vorgestellte Ansatz wird momentan in einem Projekt zur automatischen Erzeugung von Mensch-Maschine-Schnittstellen aus den Engineeringdaten eines Rohr- und Instrumentierungsdiagramms verwendet. Dabei folgt es dem autoHMI Konzept von Obst et. al. [Obs+12]. Ein weiterer Anwendungsfall bietet sich beispielsweise in der Konvertierung von verschiedenen Customizing von CAE-Systemen an. Die ersten Ergebnisse zeigen eine gute Skalierbarkeit bezüglich der Größe des Datensatzes und auch der Komplexität des Mappings. Ein weiterer Vorteil liegt darin, dass nur eine formale Sprache erlernt werden muss, um Abfragen und Transformationen durchzuführen. Dies reduziert den Einarbeitungsaufwand für neue Nutzer.

### 6.7 RDF UML Diagramm

Für die Entwicklung eines komplexen Informationssystems ist eine schnelle und einfache Kontrolle der von dem System genutzten und erstellten Daten notwendig, um eine ausreichende Qualität des Systemverhaltens zu erreichen und Aufwand bei der Fehlersuche zu minimieren. Der erste Schritt ist dabei eine nutzerfreundliche Visualisierung der verwendeten oder erstellten Datensätze. Für diese Aufgabe gibt es bereits eine Reihe von Tools. Diese sind üblicherweise auf ganz spezielle Anwendungsfälle optimiert und können Informationen, die anderen Ontologien folgen, nicht geeignet darstellen.

Auf der anderen Seite gibt es auch generische RDF-Visualisierungsansätze. Diese stellen das RDF-Basismodell im Normalfall als Graph aus Knoten und Kanten dar, in dem z.B. Navigation in der Nachbarschaft oder die Verbindung von zwei Knoten visualisiert werden können (z. B. Relfinder<sup>15</sup> oder RDF Gravity<sup>16</sup>).

Eine bisher nicht genutzte Variante liegt in der Nutzung von UML-Diagrammen zur Darstellung von RDF-Modellen. Vorteile von UML (Unified Modeling Language) [Kec11] sind eine weite Verbreitung und einfache Verständlichkeit der Diagramme. Die Visualisierungen sind zudem auch kompakt gestaltet, so dass viele Informationseinheiten übersichtlich dargestellt werden können.

Für den Anwendungsfall der Darstellung von Daten- und Informationsmodellen eignen sich aus der Fülle der UML-Diagramme das Objektdiagramm

---

<sup>15</sup><http://www.visualdataweb.org/relfinder.php>

<sup>16</sup><http://semweb.salzburgresearch.at/apps/rdf-gravity/>



und das Klassendiagramm. Diese können prinzipiell auf Linked Data übertragen werden, da die meisten Modelle hier auch einem objektorientierten Ansatz folgen.

### 6.7.1 Reine RDF Visualisierung

Üblicherweise wird ein RDF-Datensatz durch eine direkte Abbildung des Graphen auf eine Knoten- und Kantendarstellung visualisiert. Häufig wird dabei noch zwischen Ressourcen (als Kreise) und Literalen (als Rechtecke) unterschieden. Eine typische Visualisierung davon ist in Abbildung B.5 im Anhang zu sehen, die den Datensatz aus Listing B.1 visualisiert. Diese Art der Visualisierung kann einfache Sachverhalte gut darstellen, weil es die Struktur von RDF sehr gut wiedergibt. Für komplexere Informationen ist diese Form nicht mehr geeignet, so dass andere Varianten in Betracht gezogen werden müssen.

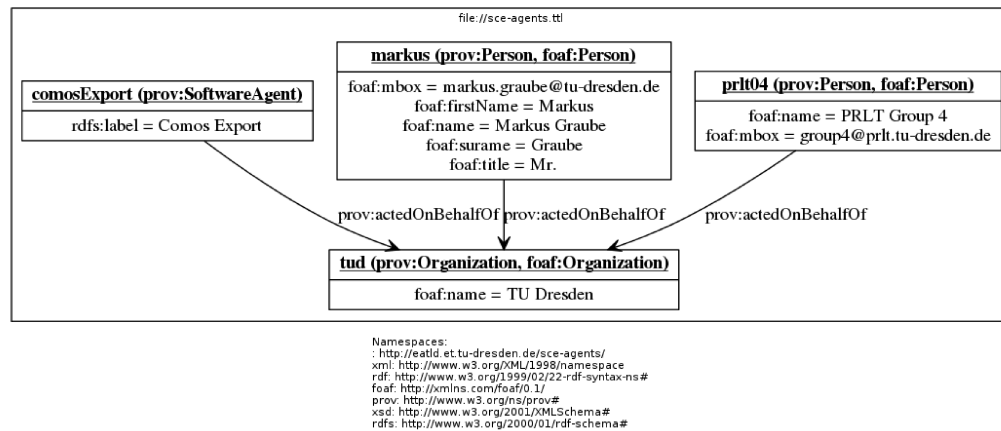
### 6.7.2 Objektdiagramm

Ein *UML-Objektdiagramm* visualisiert alle Objekte, ihre Typen, ihre Attribute und ihre Beziehungen zueinander zusammen in einem einzigen Diagramm. Objekte werden dabei als Rechteck repräsentiert, in dem oben der Name des Objekts und deren Typ dargestellt sind. Darunter sind in dem Rechteck alle Attribute des Objekts mit den aktuellen Werten aufgelistet. Die Beziehung zu anderen Objekten zeigt eine gerichtete benannte Kante an.

Abbildung 6.6 zeigt ein RDF Objektdiagramm, welches die Hauptprinzipien von UML-Diagrammen auf einen RDF Datensatz übertragen hat. Es zeigt dabei dieselben Informationen wie Abbildung B.5. Dabei wurden die Informationen jedoch komprimiert, so dass sich mehr Informationen auf einmal aufnehmen lassen und somit das Verständnis der Daten erleichtert wird.

Im Gegensatz zu einem reinen UML-Diagramm wird der Objekttyp hier durch Klammern vom Objektnamen getrennt, da der Doppelpunkt bereits der Trennung der Präfixe vorbehalten ist. Zusätzlich wurde eine Box mit den Namespaces unter dem Diagramm hinzugefügt.

Dieser Diagrammtyp unterstützt mehrere Graphen. Alle Knoten aus einem Graphen werden dabei durch ein weiteres Rechteck zusammengehalten, das zudem eine Beschreibung des Graphen enthält (`file://sce-agents.ttl` in dem Beispiel aus Abbildung 6.6). Damit können auch Datensatz-übergreifende Verknüpfungen sinnvoll dargestellt werden. Diese treten üblicherweise bei dem Einlesen mehrerer Dateien auf, können aber auch in einer Datei vorkommen,



**Abbildung 6.6:** RDF-UML-Objektdiagramm

wenn diese die Tripel im Quad-Format abgespeichert hat. In solchen Dateien (z. B. N-Quads) wird zu jedem Tripel die Zugehörigkeit zu einem Named Graph abgelegt.

Wie man in Abbildung B.6 im Anhang sehen kann werden ebenfalls größere Mengen an Tripeln (73 im Beispiel) und Named Graphs in einer übersichtlichen Form dargestellt. Für größere Diagramme sind dann jedoch auch Such-, Filter- und Navigationsmechanismen notwendig.

### 6.7.3 Klassendiagramm

Ein *UML-Klassendiagramm* ähnelt stark dem Objektdiagramm, repräsentiert aber die Klassenstruktur und die Beziehungen der Klassen untereinander. Diese werden ebenfalls in Rechtecken mit dem Klassennamen im Kopf und einer Liste der Attribute der Klasse dargestellt. Beziehungen zwischen Klassen werden mit verschiedenen Typen von gerichteten Kanten dargestellt. So wird eine Vererbungsbeziehung durch ein Kante mit offenem Pfeil am Ende repräsentiert.

Abbildung 6.7 zeigt ein beispielhaftes Diagramm der Überführung der Konzepte auf das Semantic Web. Als Klassen werden alle Ressourcen vom Typ *rdfs:Class* oder *owl:Class* erkannt. Beziehungen zwischen Klassen und Klassenattributen können durch die Properties *rdfs:domain* und *rdfs:range* der entsprechenden Properties vom Typ *owl:ObjectTypeProperty* und *owl:DataTypeProperty* abgeleitet werden.

Obwohl UML die Zusammenführung von Klassen und Instanzen in einem

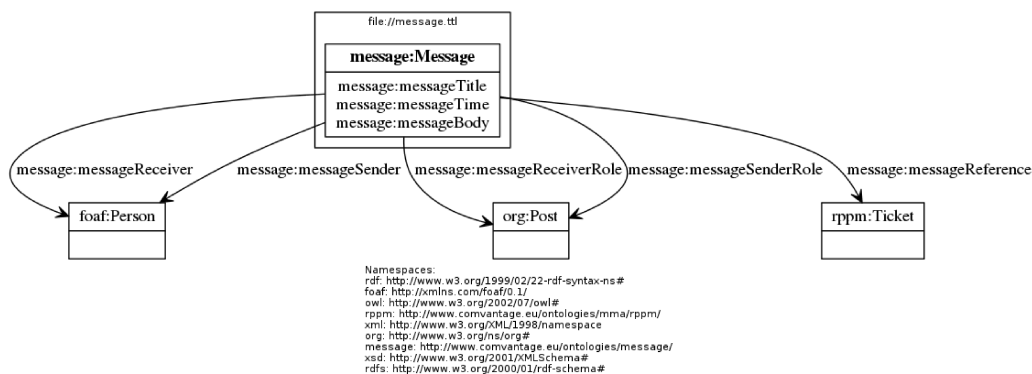


Abbildung 6.7: RDF-UML-Klassendiagramm

Diagramm eingeführt hat, bleibt dies in dem meisten Fällen und für eine automatische Visualisierung von RDF Datensätzen im Besonderen nicht gut nutzbar.

#### 6.7.4 Umsetzung

Die Visualisierungen werden durch ein Python-Skript erstellt. Der Quellcode steht unter EUPL frei und öffentlich verfügbar bei GitHub unter <https://github.com/plt-tud/rdf-uml-diagram>.

Das Skript liest ein oder mehrere RDF Dateien ein und überführt diese in einen lokalen RDF Datensatz mit verschiedenen Graphen. Über mehrere SPARQL-Abfragen auf diesen Datensatz werden die Visualisierungsobjekte erzeugt und als GraphViz-Modell abgelegt. Aus diesem Modell werden dann mit den bei GraphViz enthaltenen Layoutalgorithmen Grafiken (PNG oder SVG) erzeugt.

Prinzipiell kann derselbe Algorithmus auch direkt auf einer SPARQL-Schnittstelle eines Triplestores mit mehreren Named Graphs arbeiten. Diese Anpassung des Skripts würde dann eine breitere Nutzung des Tools ermöglichen.

Die Serialisierung als SVG ermöglicht das Einfügen von HTML-Links in die entsprechenden Knoten-Beschreibungen und bietet die Grundlage für weitergehende Navigationsansätze. Diese können durch entsprechende JavaScript-Bibliotheken realisiert werden. Sie können dann auf das SVG-DOM als Datenmodell zurückgreifen, um diesen dynamisch zu filtern, Details anzuzeigen oder zu restrukturieren.



## 7 Kritische Reflexion

### Aufbau und Einordnung des Kapitels

Gegenstand der folgenden Ausführungen ist eine kritische Reflexion der vorliegenden Arbeit. Dazu wird zunächst die Methodik dieser Arbeit diskutiert und bewertet. Anschließend wird die Frage geklärt, inwieweit mittels des LED-Ansatzes erzeugte Informationsräume den gestiegenen Anforderungen gerecht werden können und der Aufbau und Betrieb von organisationsübergreifenden Informationsräumen unterstützt werden kann. Damit sollen die anfangs aufgestellten Thesen verifiziert werden.

### 7.1 Bewertung der Methodik

Die Sammlung der Anforderungen birgt die Gefahr, dass wichtige Quellen außen vorgelassen werden. Diesem Risiko wurde durch eine strukturierte Quellenrecherche begegnet. Gleichwohl erhebt die Arbeit keinen Anspruch auf Vollständigkeit der aus der Literaturrecherche gesammelten Anforderungen. Dennoch sollten die wichtigsten Anforderungen damit abgedeckt sein. Die Anforderungserhebung ist aber mit dieser Arbeit noch nicht abgeschlossen. Neue Technologien aus anderen Bereichen werden erweiterte Möglichkeiten der Integration für die Anwender aufzeigen. Diese müssen dann in technische Anforderungen an Informationsräume gespiegelt werden.

Die Anforderungen wurden für eine Reihe von Technologien bewertet. Hier konnten auf Grund der Ressourceneinschränkung nicht alle Technologien untersucht werden, so dass hier ein Feld für weitere Analysetätigkeiten offen bleibt. Die Bewertung selbst hingegen ist durch die Definition von Bewertungskriterien gut strukturiert.

In der Konzeptphase wurden für alle noch nicht gelösten Anforderungen Konzepte ausgearbeitet. Bereits durch Linked Data angeschnittene Anforderungen wurden ebenso vorgestellt und deren Erfüllung dargelegt. Anschließend wurden die erweiterten Konzepte einzeln prototypisch implementiert. Dabei bleiben, wie im vorherigen Kapitel dargelegt, einige Punkte offen. Desweiteren ist der Reifegrad der einzelnen Komponenten nicht hoch genug, um mit dem Gesamtsystem in den Produktiveinsatz zu gehen. Dies stellt jedoch nicht das

Gesamtsystem in seiner Gesamtheit in Frage.

### 7.2 Bewertung der Ergebnisse

Das Vorgehen in dieser Arbeit konnte eine positive Beantwortung der in Kapitel 1.2 aufgestellten Hypothesen ermöglichen. Im Folgendem werden die Ergebnisse zur Bestätigung der Einzelthese nacheinander zusammengefasst.

**Neue Anforderungen an gemeinsame Informationsräume** Durch das Aufkommen der Konzepte von Industrie 4.0, CPS und Advanced Manufacturing haben sich eine große Reihe von Autoren damit beschäftigt, wie sich diese neuen Paradigmen auswirken werden. Daher wurde eine Literaturrecherche in Positionspapieren, Forschungsartikeln, Normen und Richtlinien durchgeführt. In ihrer Gesamtheit zeigen diese, dass sich nicht nur die Arbeitsteilung ändert, sondern dass auch die zur Integration notwendigen Informationsräume neuartige Bedürfnisse nach Flexibilität, Sicherheit und Semantik erfüllen müssen. Diese neuen Anforderungen decken dabei ein großes Themenspektrum ab. Somit kann These 1 (*Informationssysteme zur Erfüllung von Szenarien für Industrie 4.0, CPS bzw. Advanced Manufacturing stellen neue Anforderungen an gemeinsame Informationsräume*) als bestätigt angesehen werden.

**Unterteilung der Anforderungen in die Dimensionen Beschreibungsmittel, Methoden und Werkzeuge** Die Auflistung der Anforderung zeigte, dass sich diese von ihrer Art teilweise sehr stark unterschieden. Zur besseren Strukturierung wurden die Anforderungen auf die gängigen Dimensionen des Systemengineerings nach [Sch13] aufgeteilt. Es zeigte sich, dass in jeder der Bereiche Beschreibungsmittel, Methoden und Werkzeuge neue Anforderungen aufkommen. Die Trennung der Anforderungen konnte nicht scharf und komplett eindeutig vollzogen werden, da z.B. Informationssicherheit sowohl durch Methoden als auch Werkzeuge erreicht werden kann. Idealerweise wird diese dann durch eine Kombination der beiden erreicht. Nichtsdestotrotz kann durch die Aufteilung eine bessere Bewertung der Erfüllung von Anforderungen von Technologien vorgenommen werden. Somit kann auch These 2 mit Einschränkung der teilweisen Uneindeutigkeit bestätigt werden (*Die Anforderungen für solche Informationsräume lassen sich in die Dimensionen Beschreibungsmittel, Methoden und Werkzeuge auftrennen*). Jede Anforderung wurde durch eine Definition und Kriterien zur Erfüllung spezifiziert.

**Keine Erfüllung der Anforderungen durch heutige Systeme** In Kapitel 4 wurden etablierte Ansätze und moderne Konzepte aus der Automatisierungstechnik, Verfahrenstechnik und der Informatik auf die Erfüllung der aufgestellten Anforderungen untersucht. Dazu wurden die Ziele und Grundkonzepte von Semantic Web, OPC UA, WBEM/CIM, ISO 15926, Eclipse EMF und AutomationML zunächst beschrieben und anschließend anhand der aufgestellten Kriterien gegen die Anforderungen bewertet. Dabei hat sich gezeigt, dass keine dieser Technologien diese Anforderungen vollständig abdecken können. So ergeben sich bei dem Semantic Web Lücken in der Behandlung von dynamischen Daten, bei OPC UA hingegen gibt es offene Probleme bei der Modelltransformation und Versionierung. AutomationML ist wiederum vor allem auf statische Daten aus einem engen Umfeld spezialisiert. WEB/CIM erfordert hohe Einarbeitung und passt von dem Anwendungsgebiet nur bedingt. ISO 15926 ist zwar für die geforderte Flexibilität geeignet, scheitert aber durch seine Gründlichkeit an der geforderten Agilität. Somit kann auch These 3 als bestätigt angesehen werden (*Anforderungen an Informationsräume können von heutigen Systemen nicht vollständig erfüllt werden*).

**Linked Data und OPC UA als am besten geeignete Kandidaten** Es schneiden jedoch OPC UA und Linked Data gemessen an den Anforderungen am besten ab. Beide erfüllen die meisten Anforderungen zumindest teilweise. Nur bei den Themen Modelltransformation und Revisionierung besitzen sie Nachholbedarf. Dazu kommt, dass Linked Data dynamische Daten nur schwer in den Griff bekommt und für OPC UA der statische Austausch von Daten nur schwach ausgeprägt ist. Damit haben es bisher beide nicht in den umgreifenden Einsatz in der Industrie geschafft. Dies stützt These 4 (*Linked Data erfüllt einen Großteil der gestellten Anforderungen, ist aber mit den heutigen Konzepten und Technologien industriell nicht einsetzbar*).

**Erweiterung von Linked Data zu Linked Enterprise Data** Dabei ist Linked Data gerade im Bereich der organisationsübergreifenden Integration stärker, weshalb es zur weiteren Betrachtung näher untersucht wurde. Die in Kapitel 5 beschriebenen Ergänzungen erlauben Linked Data die Anforderungen insgesamt zu erfüllen. Zu diesen Erweiterungen gehört eine Absicherung des Informationsraumes durch ein feingranulares Informationssicherheitssystem, die Anbindung an existierende industrielle Informationssysteme, eine Möglichkeit zu Revisionierung, die performante Einbindung von dynamischen

Daten, Prozesse zum Ausgleich von Flexibilität und Standardisierung sowie eine Unterstützung bei der Nachverfolgbarkeit. Somit kann auch These 5 (*Geeignete Erweiterungen im Bereich der Beschreibungsmittel, Methoden und Werkzeuge lassen daraus Linked Enterprise Data entstehen, das die erfassten Anforderungen abdeckt*) teilweise als erfüllt gelten. Die notwendigen Erweiterungen beziehen sich nämlich ausschließlich auf die Bereiche Methoden und Werkzeuge. Hingegen hat sich bei Linked Data gezeigt, dass es eine ausreichend hohe Mächtigkeit in den Beschreibungsmittel hat, die eine Erweiterung für die Erfüllung der aufgestellten Anforderungen zunächst nicht notwendig machen.

**Weitere Ergebnisse** Das vorgestellte Vorgehen zur Integration von industriellen Daten in einem flexiblen Umfeld erreicht eine zufriedenstellende Qualität und Gebrauchstauglichkeit, so dass damit bereits einige Applikationen umgesetzt werden konnten. Z.B. konnte in [ZGU12] gezeigt werden, dass eine komplexe Anwendung auf der Basis dieses Konzepts in Verbindung mit RFID signifikante Vorteile in der Ausführung von Wartungsaufträgen erbringen kann. Ebenso zeigen [UPZ11] und [Pfe+12], dass man auf der Netzwerkstruktur von RDF nutzerfreundliche Nachbarschaftserkundungswerkzeuge gestalten kann, die dem Nutzer bei dem Auffinden ihm bisher unbekannter Informationen unterstützen.

Gerade bei der App-Orchestrierung [Zie+12; Pfe+13] zeigen sich die Vorteile des LED-Ansatzes [Gra+13c]. Hier müssen die Apps nicht nur für den aktuellen Business-Prozess ausgesucht werden, sondern auch an die Daten für diesen Prozess angepasst werden. Dies kann einerseits durch den einfachen Austausch der in der App genutzten SPARQL-Abfragen geschehen. Eine andere Möglichkeit ist es, die Mächtigkeit der Semantik auszunutzen und SPARQL-Abfragen zu schreiben, die so auf den Ontologien definiert sind, dass eine kleine Zusatzinformation eine Verknüpfung der neuen Daten zu den erwarteten Datenstruktur der App (abgebildet in einer weiteren Ontologie) ausreichend für die Anpassung ist.

Auch in der Industrie wird das Konzept von Linked Data untersucht. Es gibt einerseits die konzeptionelle Bereitschaft Linked Data als Informationstechnologie einzusetzen, so dass es in der Normungsroadmap Industrie 4.0 auftaucht [DD15]. Zum anderen werden auch konkrete Konzepte in dieser Richtung bereits umgesetzt. So wird beispielsweise in [Sun14] ein Konzept zur Fernwartung von intelligenten Komponenten über Linked Data Schnittstellen erstellt und implementiert.



## 8 Zusammenfassung und Ausblick

### Aufbau und Einordnung des Kapitels

In dieser Arbeit wurde der Bogen von Betrachtungen zu Anforderungen von zukünftigen Informationsräumen über momentan existierende Technologien hin zum Konzept, Umsetzung und Evaluation eines solchen Informationsraumes mit Hilfe von *Linked Enterprise Data* aufgespannt. Im nachfolgenden Kapitel sollen Fragen dahingehend beantwortet werden, wie die erreichten Ziele der Arbeit einzuordnen sind und welche zukünftigen Entwicklungen abgeleitet werden können. In Abschnitt 8.1 erfolgt zunächst ein kurzer Umriss der Arbeit. Gegenstand weiterer Ausführungen ist die Darstellung der erreichten Ergebnisse und Innovationen der Arbeit sowie bestehender Grenzen und Defizite (Abschnitt 8.2). Ein Ausblick auf mögliche Erweiterungen des LED-Ansatzes bildet den Abschluss der Arbeit (Abschnitt 8.3).

### 8.1 Zusammenfassung

Insgesamt zeigt sich, dass Linked Enterprise Data einen geeigneten Kandidaten für die Schaffung eines integrierten Informationsraumes darstellt. Dabei spielt dieser Ansatz seine Stärken vor allem dann aus, wenn ein hohes Maß an Flexibilität und Geschwindigkeit gefordert ist. Im Zusammenspiel mit OPC UA bietet es sich so als gute Technologie an, um Integration auf höheren Ebenen zu erreichen. Dabei baut es auf den Grundkonzepten von Linked Data auf und wird um Komponenten und Konzepte erweitert, die es erlauben, die aufgestellten Anforderungen zu erfüllen.

### 8.2 Innovationen und Grenzen

Die größten Vorteile, die sich durch LED ergeben, sind die Nutzung von vielen vorhanden Informationshaushalten aus der Linked Open Data Bewegung sowie die Flexibilität und Agilität, die aus den Linked Data Konzepten resultiert. LED ist ein offenes Konzept, das sich leicht erweitern und auf spezielle Anforderungen anpassen lässt. Durch die semantische Integration auf Datenebene behält LED auch bei individuellen Erweiterungen seine Kompatibilität

zwischen verschiedenen Versionen. Die Integration von statischen Informationen über Unternehmensgrenzen hinweg auf eine semantische Art und Weise ist ein starkes Alleinstellungsmerkmal gegenüber anderen Technologien.

Mit der Erweiterung durch R43ples konnte eindrucksvoll gezeigt werden, dass eine Versionsverwaltung nahezu nahtlos in LED mit einem SPARQL-Zugriff integriert werden kann. Dabei ist das System komplett semantisch beschrieben, so dass die Wahrheit über die Revisionen nicht im Code des Werkzeugs, sondern in der Semantik der Daten liegt und somit auch von anderen Werkzeugen ohne explizites Wissen genutzt werden kann.

Die größte Leistungssteigerung bei R43ples konnte durch ein Query-Rewriting erreicht werden. Der Ansatz, eine Query umzuschreiben, um Informationen zu filtern, transformieren oder anders darzustellen, hat sich hier ebenso wie beim SPARQL Security Query Rewriting als erfolgreich und mächtig herausgestellt. Wo bei diesen beiden Ansätzen die Transformation der Query noch durch Programme in Java erfolgte, wurde bei der SPARQL Model Transformation noch einen Schritt weitergegangen. Hier wird die Query selbst als Modell behandelt, das auch in Linked Data dargestellt wird. Diese Query in RDF kann nun wiederum durch SPARQL als Transformationsengine so bearbeitet werden, dass z.B. aus deklarativen TGG-Regeln operationale TGG-Regeln erzeugt werden können.

Auf Grund der Entscheidung, auf den etablierten Internettechnologien aufzubauen, kann LED für durchsatzstarke und latenzkritische Kommunikation nicht direkt eingesetzt werden. Der Einsatz von HTTP mit dem kompletten Verbindungsaufbau bei jeder Kommunikation unterbindet solch ein Echtzeitverhalten. Die Entwicklung von echtzeitkritische Anwendungen ist auch weiterhin auf die spezialisierten Lösungen der Automatisierungstechnik angewiesen.

Momentan laufen in der Automatisierungstechnik viele Entwicklungen auf OPC UA hinaus. Wie bereits gezeigt, kann hier eine gute Integration dieser beiden Technologien gelingen, so dass sich LED um die Integration der Informationseinheiten auf höheren Ebenen kümmert während OPC UA auf den unteren Ebenen der Automatisierungspyramide zu finden sein kann. Dies stützt die These, dass vor allem ein Zusammenspiel von OPC UA und LED vielversprechend ist, wenn beide Technologien ihre Stärken ausspielen können. Wenn es nun für OPC UA noch geschafft wird, Echtzeitfähigkeitseigenschaften in das Protokoll zu integrieren, dann kann OPC UA sogar auf der Feldebene als einziger Kanal eingesetzt werden, so dass eine komplette semantische Integration von den Sensoren bis zu der Beschreibung von Unternehmenseinheiten erfolgen kann.

## 8.3 Ausblick

In dieser Arbeit wurden viele Komponenten vorgestellt, die zusammen das Konzept Linked Enterprise Data bilden. Allen diesen Komponenten ist gemein, dass sie im Rahmen dieser Arbeit nur prototypisch umgesetzt wurden, so dass überall ein Weiterentwicklungsbedarf vorhanden ist. Im Folgenden werden wichtige Weiterentwicklungsmöglichkeiten sowie neue Forschungsrichtungen aufgezeigt.

**R43ples** R43ples hat mittlerweile einen hohen Reifegrad als Werkzeug für die Revisionierung in semantischen Netzwerken erreicht. Offen bleibt aber noch die Fragestellung nach einer verteilten Versionsverwaltung, in der Teile der Daten und Revisionen auf unterschiedlichen Servern gespeichert wird. Auch im Bereich der Performance gibt es noch Potential, das z.B. durch Mechanismen, die ein intelligentes Vorhalten von bestimmten Revisionen erlauben, ausgereizt werden kann. Ebenso fehlt R43ples noch eine semantische Autorisierungskomponente, die verschiedenen Nutzern und Rollen Rechte für Aktionen auf bestimmten Graphen erlauben kann. Eine andere Weiterentwicklungsmöglichkeit liegt in der Übertragung der Konzepte auf OPC UA, um auch dort das vorhandene, aber schlecht verwendbare Versionierungskonzept durch eine semantische Beschreibung zu erweitern.

**SPARQL Security** Es konnte gezeigt werden, dass der SPARQL Security Rewriting Ansatz für Named Graphs und SPARQL funktioniert. Nächste Schritte sind die Zusammenführung mit einer stabilen und zuverlässigen Authentifizierungskomponente sowie die komplette Abdeckung des SPARQL 1.1 Sprachumfangs.

**Design For Evolution und Modelltransformation** Obwohl LED einen großen Grad an Freiheit und Änderbarkeit aufweist, ist es dennoch notwendig, sich intensiv damit zu beschäftigen, wie diese Möglichkeiten sinnvoll genutzt werden können und sollten. Offene Fragen sind hier, wie Modelle so gestaltet werden können, dass sie einfach an äußere Änderungen angepasst werden können. Weiterhin ist die Gestaltung von Systemen zu klären, deren Komponenten sich selbst ändern (Ersetzungen, Aktualisierungen). Hier muss das Konzept einer wirklich verteilten Revisionierung angegangen werden, dass nicht nur Daten, sondern auch Werkzeuge und deren Kompatibilitäten betrachtet. Dabei spielt auch das Thema Modell-Transformation eine wichtige

Rolle, so dass hier eine Anwendung der SPARQL-Modell-Transformation (SMT) mit TGG für Roundtrip-Engineering zu finden ist. Mit dieser kann gleich der Praxisbezug an einem Beispiel nachgewiesen werden.

**Query Transformation** Die semantische Integration von Daten aus unterschiedlichen Quellen wurde in dieser Arbeit durch Modelltransformationen gelöst, die hauptsächlich auf Zusammenhängen der Meta-Modelle aufbauen. Daraus ergeben sich für die Datensätze unterschiedliche Sichten, die jeweils einem Informationsmodell entsprechen. Dieser Ansatz kann dahin erweitert werden, dass nun nicht nur Daten zwischen verschiedenen Informationsmodellen ineinander überführt werden können, sondern auch die Abfragen auf diese Daten. Dies würde es einem Client erlauben, verschiedene Endpoint jeweils in seiner bevorzugten Semantik abzufragen und die Ergebnisse auch genau in dieser zu erhalten ohne sich Gedanken darum machen zu müssen, ob die Server diese Semantik auch verstehen. Die Server hingegen halten die Daten nun nicht mehr unbedingt in allen Semantiken vor, sondern übersetzen die Query vor der Abfrage in ihre native Semantik. Dieser Ansatz der *Query-Transformation* ist in Abbildung B.10 im Anhang dargestellt. Eine App als Client kann seine Query  $Q$  in seiner gewählten Semantik  $M1$  ab drei Endpunkte schicken. Die Endpunkte übersetzen nun die Query durch  $QT1$ ,  $QT2$  und  $QT3$  in eine Semantik, die der jeweilige Endpoint versteht und führen sie dann auf den Daten aus. Dazu benötigen die Query-Transformatoren die Transformationsregeln  $T13$ ,  $T12$  und  $T23$  zwischen den Meta-Modellen  $M1$ ,  $M2$  und  $M3$ .

Durch diesen Ansatz könnten Queries in einem selbst gewählten Modell formuliert werden. Die Query könnte dann an einen beliebigen Server geschickt werden, der die Query automatisch in das eigene Modell übersetzt und ausführt sowie die Ergebnisse wieder in das für die Abfrage genutzte Modell zurück transformiert. Hierbei können natürlich auch Informationsverluste auftreten, wenn die Transformation nicht eindeutig formuliert werden kann. Der Vorteil ist, dass hier eine komplett dezentrale Architektur genutzt werden kann, bei dem sich auf kein Referenzmodell geeinigt werden muss. Nichtsdestotrotz bleibt jedoch eine Einigung auf grundlegende Konzepte eines Referenzmodells sinnvoll, die jeder Partner für sich implementieren kann. ( $n \cdot (n-1)$  Transformationsmodelle mit SKOS, OWL)

Eine Möglichkeit, solche oder andere sinnvolle Transformationen zu beschreiben und durchzuführen, bietet SPARQL an. Hierbei besteht eine weitere offene Fragestellung darin zu untersuchen, inwieweit solche SPARQL-

Transformationen (teil-)automatisch aus bestehenden Ontologien und deren Verknüpfungen erzeugt werden können. Hier sind z.B. die Properties *owl:sameAs*, *rdfs:subPropertyOf*, *rdfs:subClassOf* sehr interessant. Weiterhin erscheinen die Konzepte aus SKOS vielversprechend, da sie eine große Mächtigkeit der Ausdrucksfähigkeit für diesen Anwendungszweck besitzen.

**OPC UA SPARQL Endpunkt** Eine stärkere Kopplung von OPC UA und LED sollte dadurch erreicht werden, dass die Vorteile einer deklarativen und mächtigen Abfragesprache wie SPARQL auf das ebenfalls graphenorientierte Namensraummodell von OPC UA übertragen werden. Dies würde einerseits die Mächtigkeit von OPC UA selbst erheblich stärken durch eine Integration als eigenen erweiteren Query-Dienst in OPC UA. Zum anderen werden damit dann komplexe und schnelle Abfragen möglich, die sowohl auf den organisationsübergreifenden Daten als auch den lokalen hoch-performanten OPC UA Daten arbeiten.



# Anhang





## **A Kriterien für die Anforderungsbewertung von Informationsräumen**

In den folgenden Tabellen A.1, A.2 und A.3 sind die Kriterien für die Erfüllung der Anforderungen an Informationsräume beschrieben, wie sie in Kapitel 3 genutzt werden. Dabei sind die Kriterien aufgeführt, die ein Informationsraum erfüllen muss, um die jeweilige Anforderungskategorie nicht, teilweise oder vollständig zu erfüllen.

**Tabelle A.1:** Kriterien hinsichtlich der Erfüllung der Anforderungen im Bereich Beschreibungsmittel

		keine (✗)	teilweise (✓)	vollständig (✓✓)
B1)	Mächtigkeit der Informationsmodellierung	Nur Basiskonzepte (Attribute, Merkmale, Variablen, einfache Datentypen)	Objektorientierung	Vererbung, Referenzen, Komplexe Datentypen
B2)	Semantik	Vordefinierte Konzepte	Nachschlagbare maschinenlesbare Konzepte	Konzepte im selben Informationsraum
B3)	Erweiterbarkeit	feste Datentypen und Objekttypen	Objekttypen erweiterbar	IM auf Erweiterbarkeit ausgelegt
B4)	Selbstbeschreibungsfähigkeit	Modelle und Werkzeuge können sich nicht selbst beschreiben	Modelle und Werkzeuge können über eigene Eigenschaften und Fähigkeiten Auskunft geben	Modelle und Werkzeuge können über eigene Eigenschaften und Fähigkeiten semantische Auskünfte geben
B5)	Verknüpfungen	Keine Festlegung	Lokale Verknüpfungen	Standardmäßige Nutzung von organisationsübergreifenden Verknüpfungen

**Tabelle A.2:** Kriterien hinsichtlich Anforderungsbewertung im Bereich Methoden

		keine (✗)	teilweise (✓)	vollständig (✓✓)
M1)	Abstraktion der Kommunikation	Keine Festlegung der ISO/OSI-Schichten 1-4	Spezifische Protokolle für Schichten 1-4	Nutzung etablierter Protokolle für Schichten 1-4
M2)	Statische Daten	Keine Unterstützung	Persistierung von Daten möglich	Standardisierte Austauschformate
M3)	Dynamische Daten	Keine Unterstützung	Nutzung einer Datenbank mit dynamischer Schnittstellen	Hochdynamischehaltung und Zugriff
M4)	Informationssicherheit	Keine Unterstützung	Einfache Autorisierungsmechanismen	Verschiedene Views auf Informationen
M5)	Dienste-Orientierung	Keine Unterstützung von Diensten	Vorgefertigte Dienste um Information über Netzwerk abrufen zu können	Frei definierbare Dienste über definierte Schnittstellen

**Tabelle A.3:** Kriterien hinsichtlich Anforderungsbewertung im Bereich Werkzeuge

		keine (✗)	teilweise (✓)	vollständig (✓✓)
W1)	Modelltransformationen	Keine Unterstützung	Existierende Werkzeuge zur Modelltransformation	Integrierte Modelltransformationskonzepte
W2)	Revisionierung	Keine Unterstützung	Versionsinformationen können abgelegt werden	Mechanismen um auf alte Revisionen zuzugreifen
W3)	Entwicklungsunterstützung	Keine nennenswerten Werkzeuge verfügbar	Werkzeuge für die Modellierung, Analyse, Erstellung verfügbar	Freie und offene Werkzeuge verfügbar
W4)	Schnittstellen/Adapter für andere Systeme	Keine Adapter verfügbar	Einzelne Adapter verfügbar	Adapter für eine große Anzahl an Systemen verfügbar
W5)	Inspizierbarkeit zur Laufzeit	Keine Unterstützung	Einfache Werkzeuge zur Inspektion	Graphische Werkzeuge vorhanden

## **B Abbildungen und Listings**

Dieses Kapitel führt einige Abbildungen und Listings zu Details der Implementierung auf.

**Listing B.1:** Beispielhafter RDF Datensatz in Turtle-Serialisierung mit 19 Triples

```
@prefix : <http://eatld.et.tu-dresden.de/sce-agents/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .

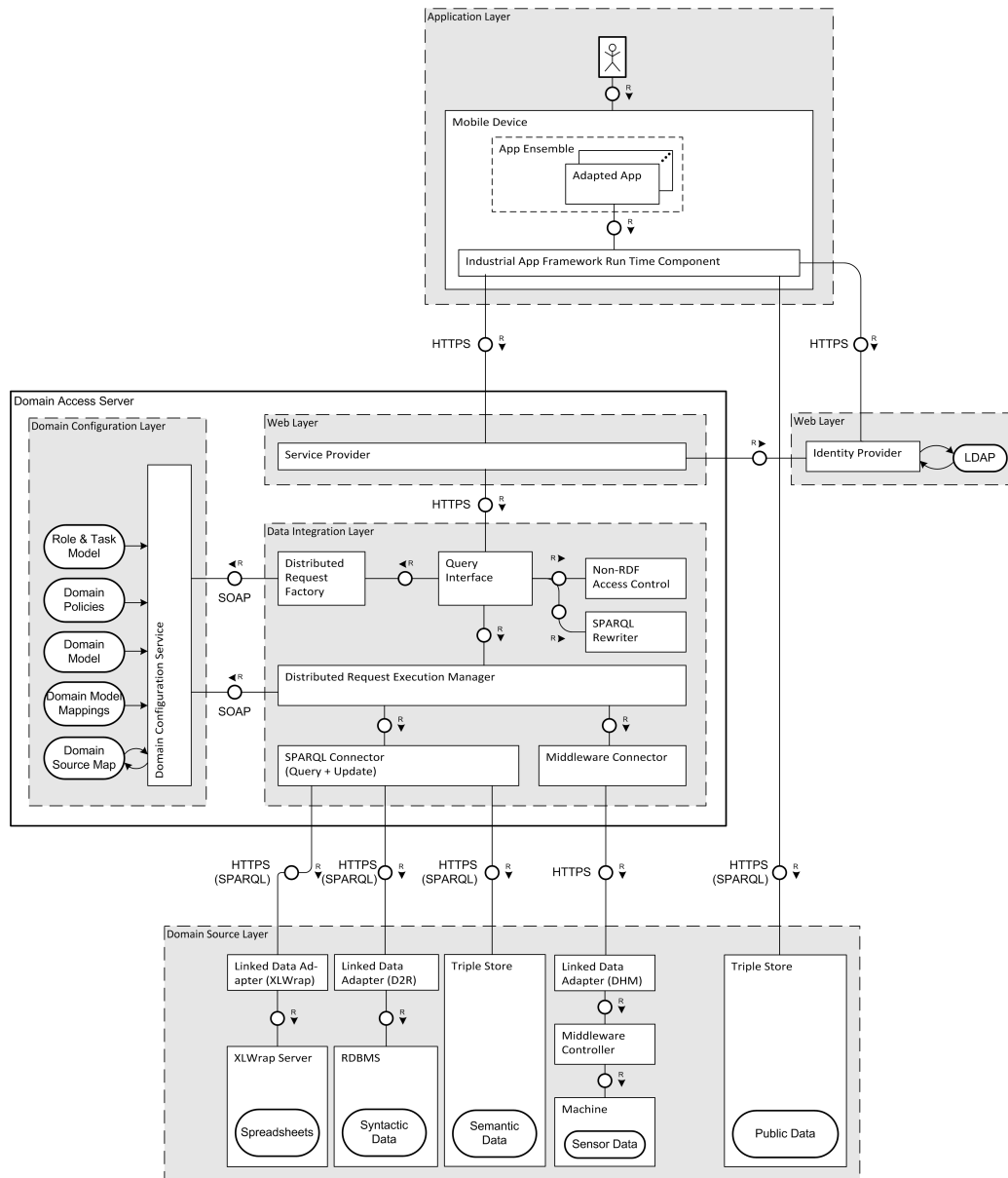
:prlt04 a foaf:Person, prov:Person;
    foaf:name "PRLT Group 4";
    foaf:mbox "group4@prlt.tu-dresden.de";
    prov:actedOnBehalfOf :tud.

:markus a foaf:Person, prov:Person;
    foaf:name "Markus Graube";
    foaf:title "Mr.";
    foaf:firstName "Markus";
    foaf:surname "Graube";
    foaf:mbox "markus.graube@tu-dresden.de";
    prov:actedOnBehalfOf :tud.

:comosExport a prov:SoftwareAgent;
    rdfs:label "Comos Export";
    prov:actedOnBehalfOf :tud.

:tud a foaf:Organization, prov:Organization;
    foaf:name "TU Dresden".
```

---



**Abbildung B.1:** Abbildung der implementierten Komponenten auf Architektur

**Listing B.2:** Informationsmodell zu Listing 6.11

---

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix ins: <http://www.comvantage.eu/instances#>.

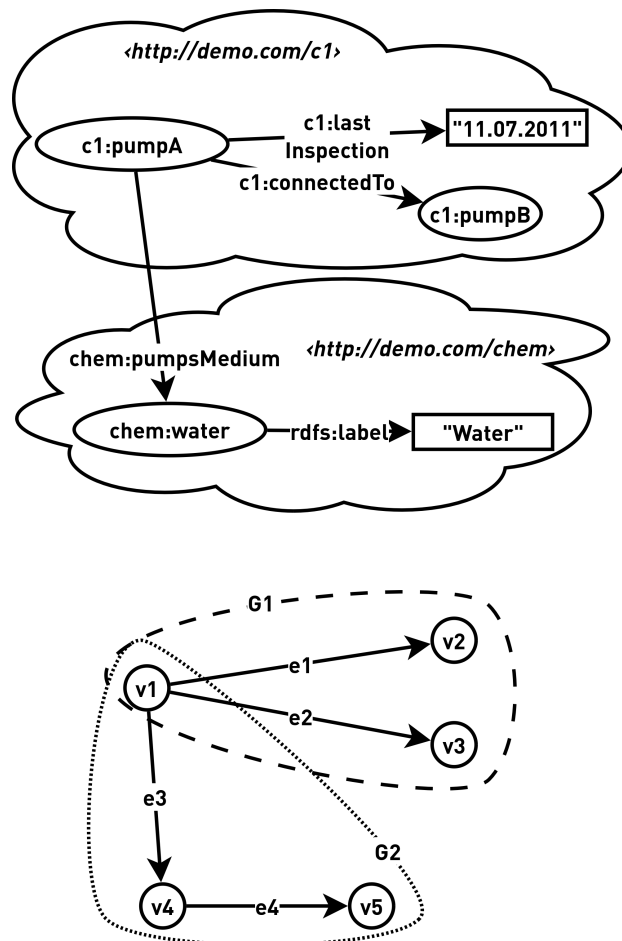
ins:jsmith a foaf:Person;
  rdfs:label "John Smith";
  foaf:firstName "John";
  foaf:lastName "Smith";
  foaf:mbox "jsmith@comau.it";
  foaf:birthday "05-28";
  foaf:knows ins:aharris, ins:pshaw.

ins:aharris a foaf:Person;
  rdfs:label "Arthur Harris";
  foaf:firstName "Arthur";
  foaf:lastName "Harris";
  foaf:mbox "aharris@dresscode21.com";
  foaf:birthday "12-22";
  foaf:knows ins:jsmith.

ins:pshaw a foaf:Person;
  rdfs:label "Peter Shaw";
  foaf:firstName "Peter";
  foaf:lastName "Shaw";
  foaf:knows ins:jsmith.
```

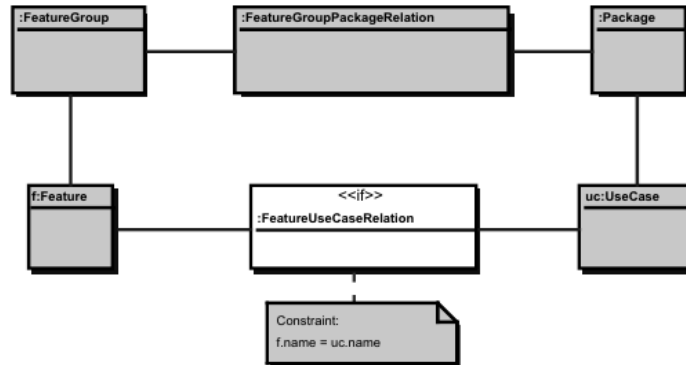
---



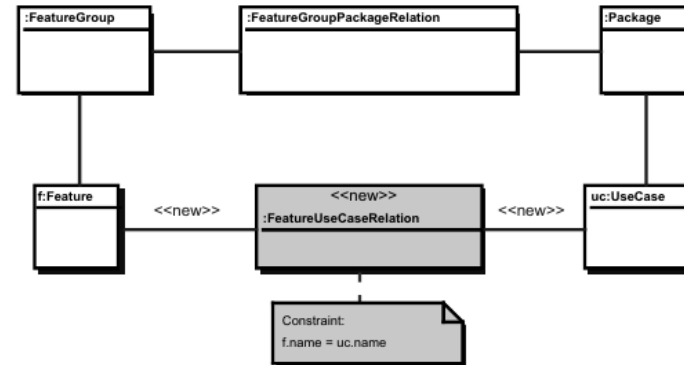


**Abbildung B.2:** Zusammenhang zwischen mathematischen Graphen und Linked Data

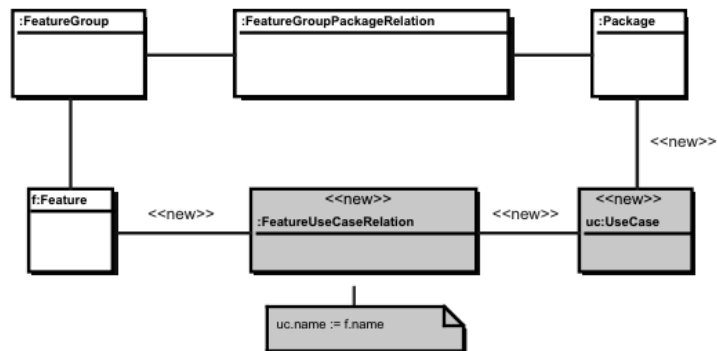
a) checkConsistency()



b) createCorrespondenceLink()



c) doLRTransformation()



d) doRLTransformation()

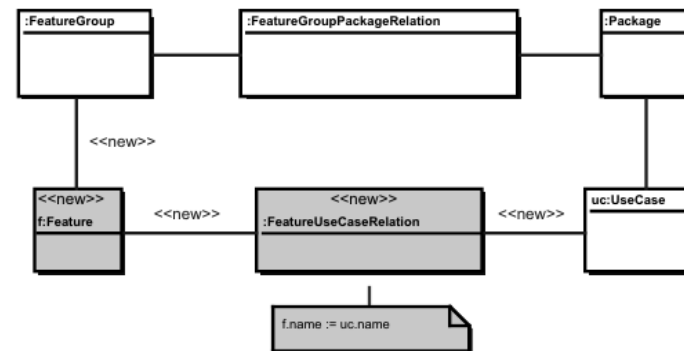
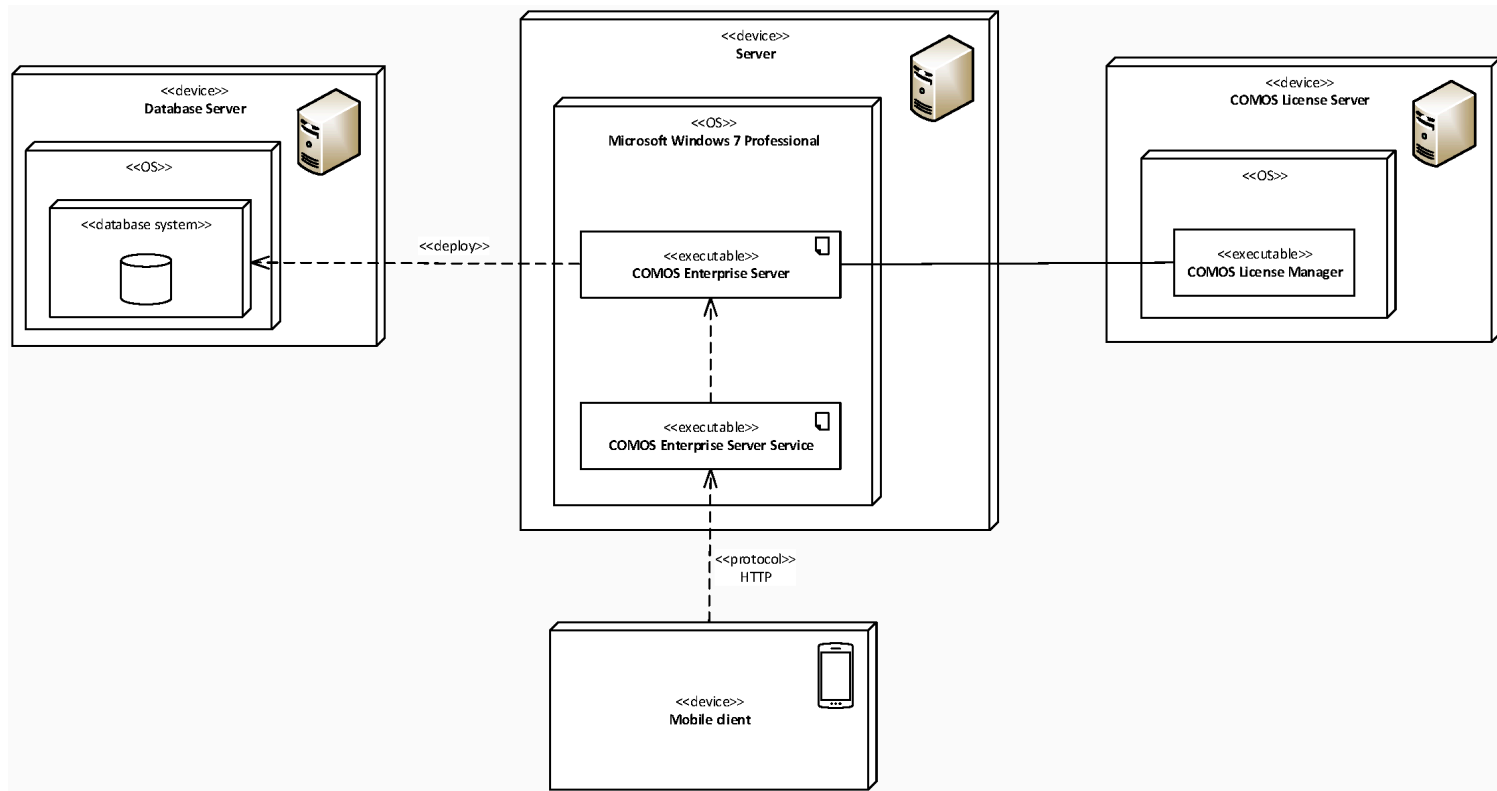
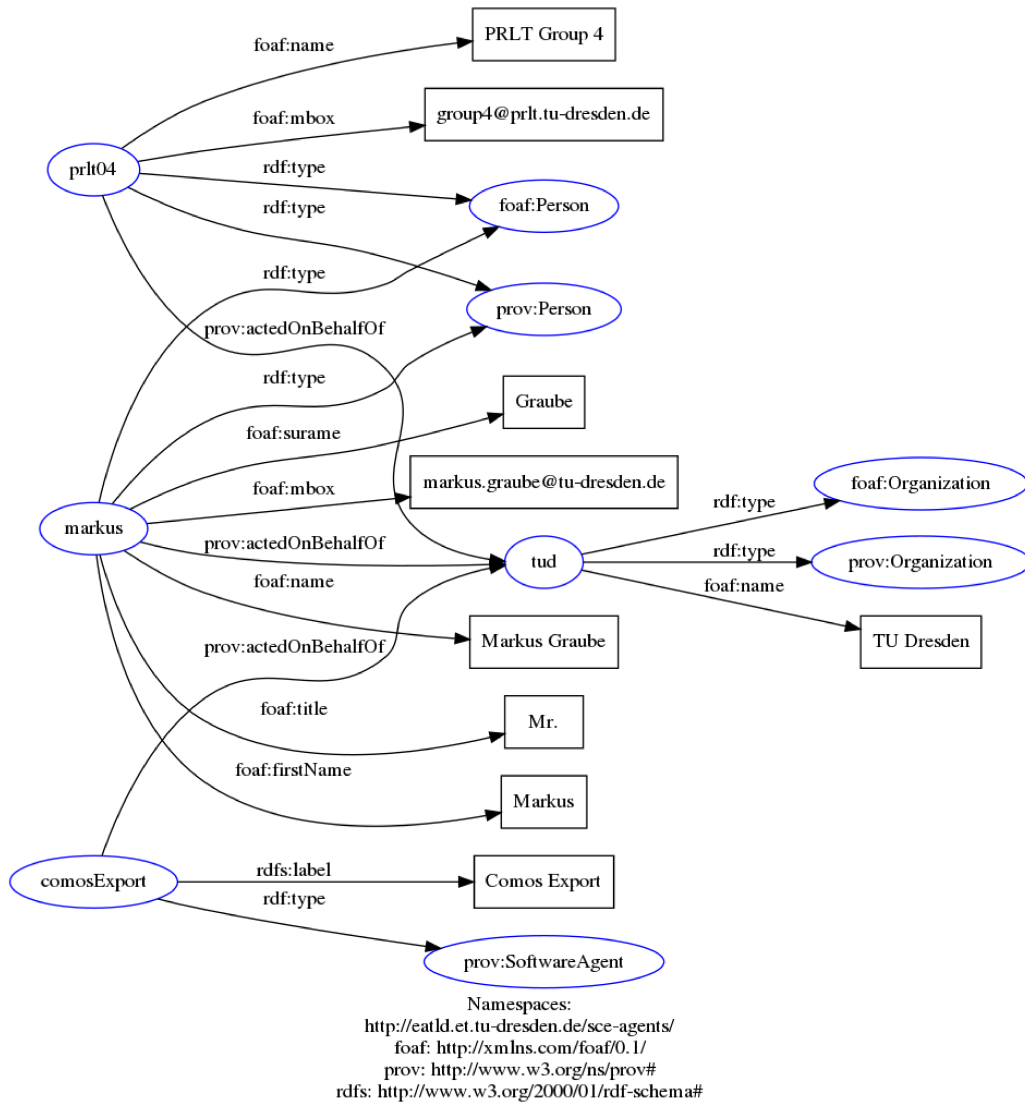


Abbildung B.3: Ableitung von operationalen Regeln aus Abb. 2.3 [KS06]



**Abbildung B.4:** CESS Verteilungsdiagramm



**Abbildung B.5:** Graphische RDF Repräsentation des Beispiel-Datensatzes aus Listing B.1

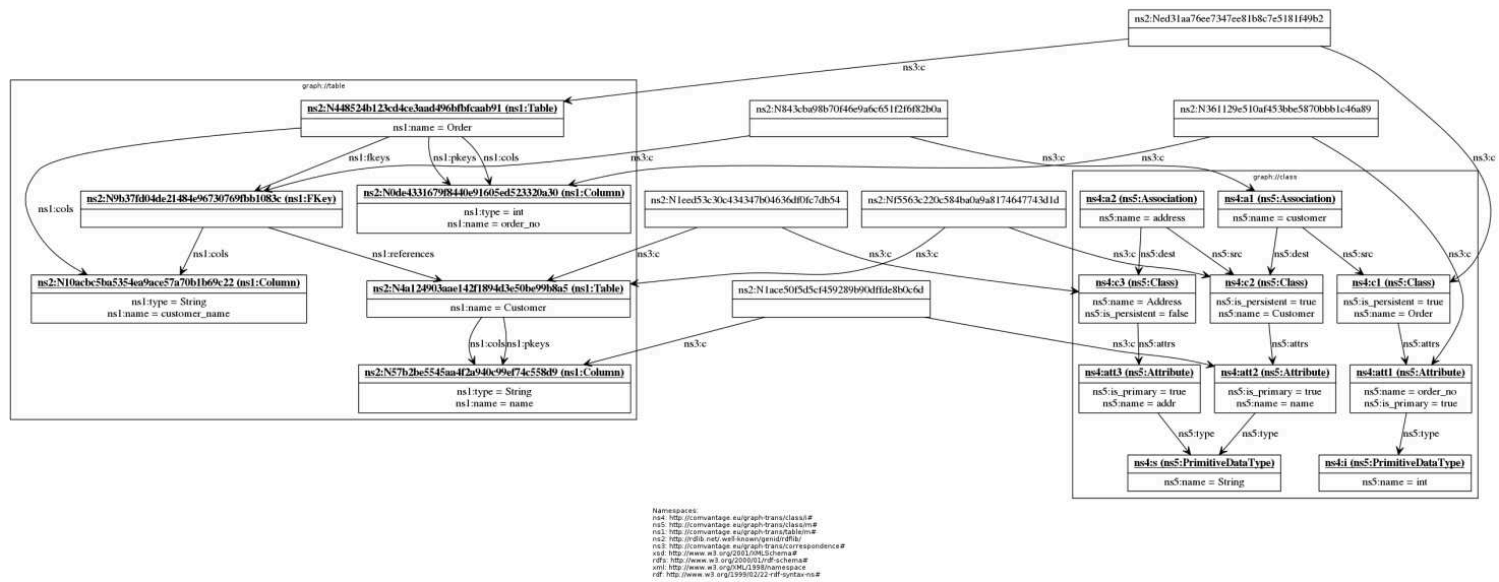
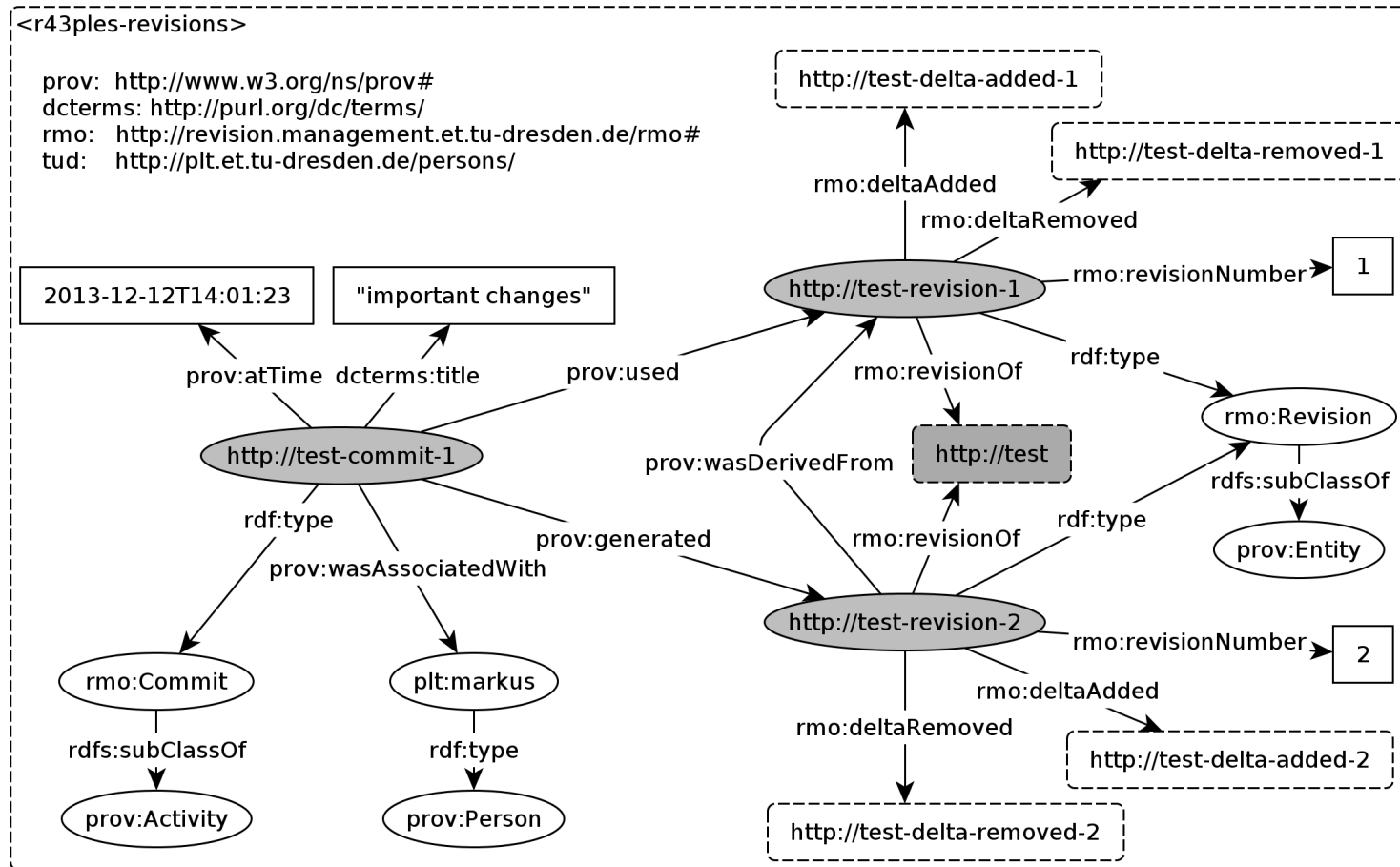



Abbildung B.6: Großes RDF-UML-Objektdiagramm mit 73 Tripeln



**Abbildung B.7:** Informationsmodell eines einfachen Revisionsgraphen in R43ples von dem Graphen <http://test> mit zwei Revisionen und den dazugehörigen Commit

 R43ples

Endpoint

Debug

Merging

Git: 3d08cba - master

[Website](#)

[GitHub](#)

### Query Form

Query Text

```
OPTION r43ples:SPARQL_JOIN
SELECT ?s ?p ?o
FROM <http://test.com/r43ples-dataset-1> REVISION "master"
FROM <http://test.com/r43ples-dataset-2> REVISION "2"
WHERE {
  ?s ?p ?o.
}
```

Results Format: HTML

JOIN Option: ☐

Run Query

Reset

### Revision Information

All

Create sample data set

(All)

Turtle

Get Revision Graph

(None)

Drop Graph

### Example Queries

Create Graph under Version Control

Select Query

Select Query II

Select Query - Multiple Graphs

copy to form

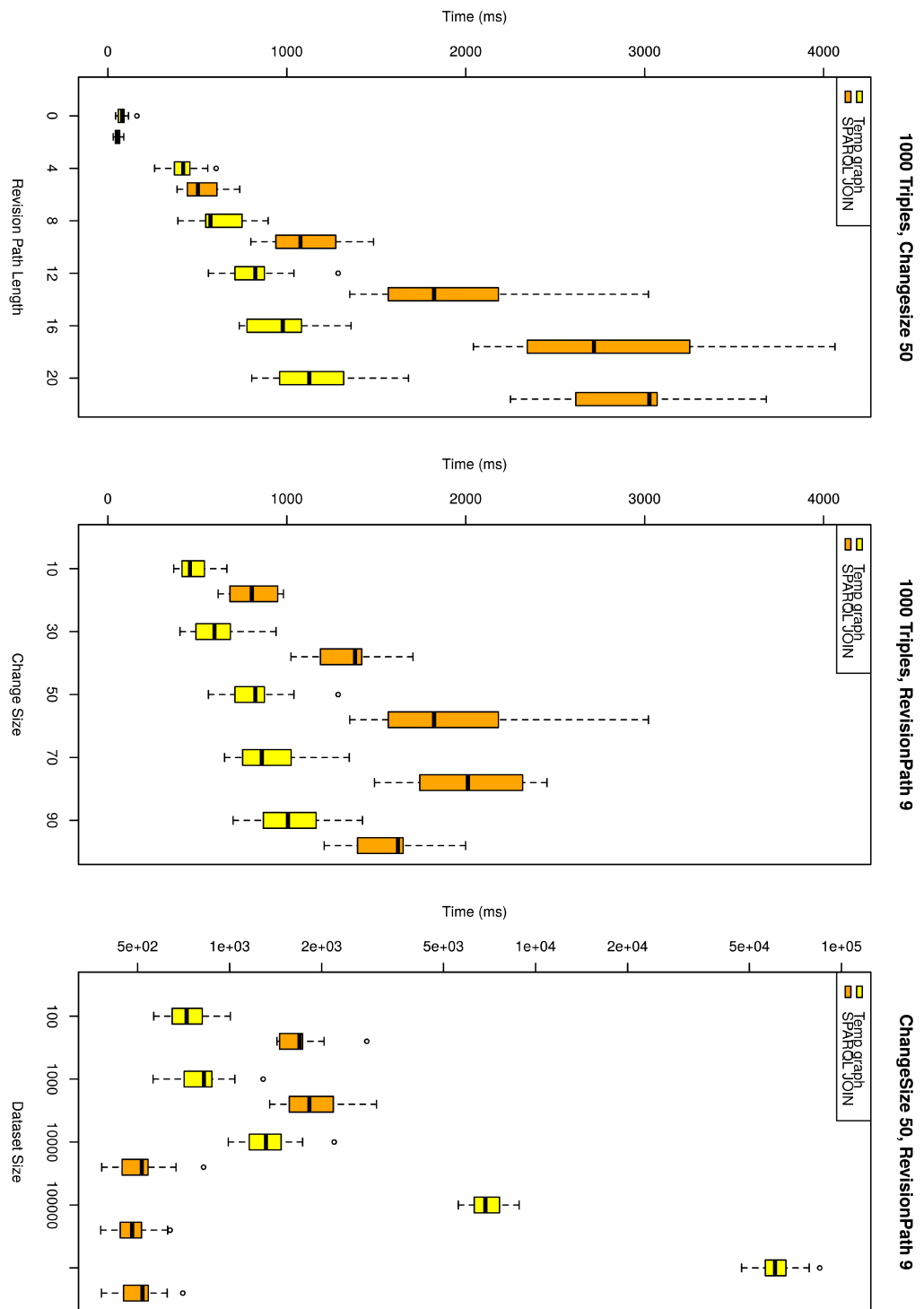
```
OPTION r43ples:SPARQL_JOIN
SELECT ?s ?p ?o
FROM <http://test.com/r43ples-dataset-1> REVISION "master"
FROM <http://test.com/r43ples-dataset-2> REVISION "2"
WHERE {
  ?s ?p ?o.
}
```

Update Query

Branching

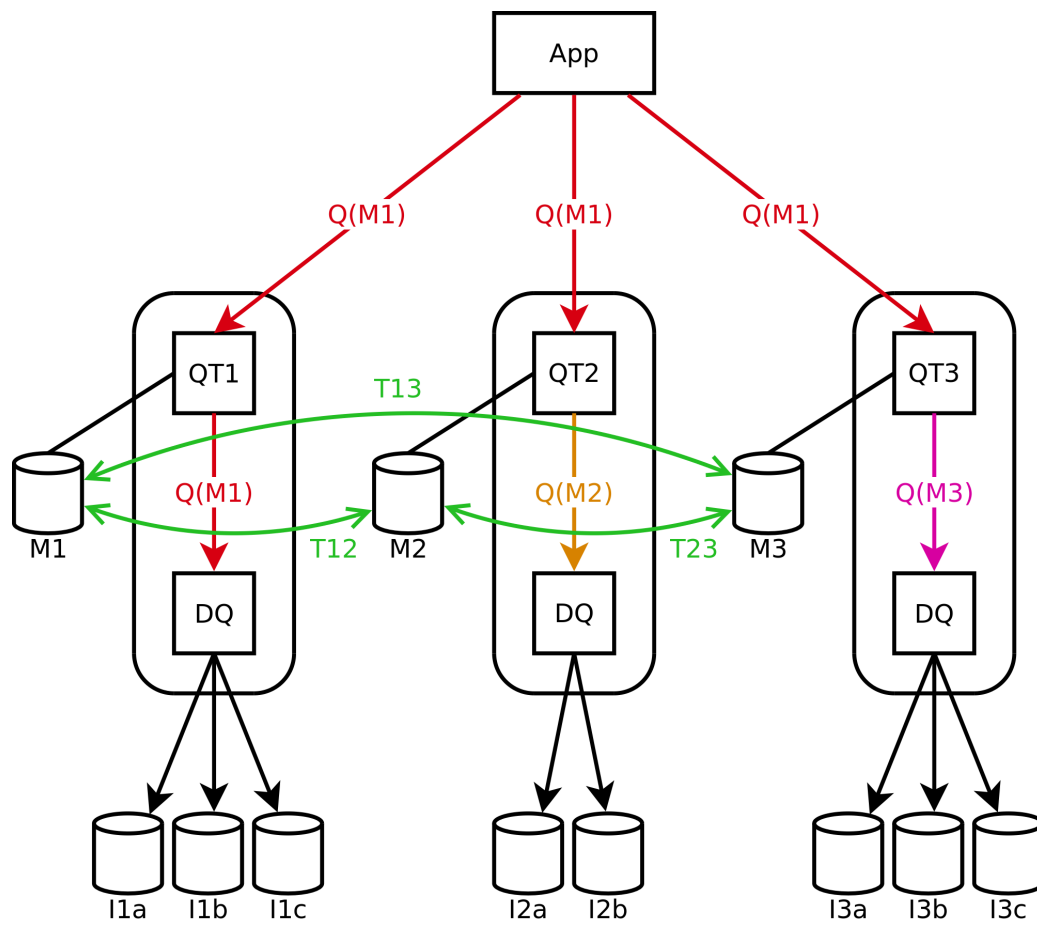
Tagging

Abbildung B.8: Screenshot von R43ples



**Abbildung B.9:** Vergleich der Antwortzeiten von R43ples zwischen der Query Rewriting Option (dunkelgrau) und der Option mit Erstellung eines temporären Graphen (hellgrau) für eine ansteigende Entfernung zur herzustellenden Revision (links), der Größe des Changesets (Mitte) und der Datensatzgröße (rechts)





**Abbildung B.10:** Konzept für Query-Transformation



## C Linked Data Grundlagen

### C.1 Technologien

Diese Ziele werden bei dem Semantic Web mit einer Reihe von Technologien gelöst, die man in ihrer Gesamtheit auch den *Semantic Web Stack* nennt. Abbildung C.1 zeigt eine mögliche Ausprägung dieses Stacks.

#### C.1.1 IRI/URI

Zur eindeutigen Identifikation von Informationsentitäten werden in Linked Data International Resource Identifiers (IRIs) genutzt. Diese sind in RFC 3987 [DS05] definiert und der Nachfolger von Uniform Resource Identifiers (URIs) mit der Unterstützung für internationale Zeichen. URIs sind in der RFC 3986 [BFM05] beschrieben und können aus einem Protokoll, einem Server oder Anbieter, einem Pfad, einer Abfrage und einem Fragment bestehen, wobei nur das Protokoll und der Pfad obligatorisch sind.

protokoll://server/pfad?query#fragment

#### C.1.2 RDF

Das Resource Description Framework (RDF) [Sch+14b] ist das verwendete Datenmodell für Linked Data. Es zeichnet sich durch seine Einfachheit und fehlende Ausdrucksstärke aus. Alle Informationen werden als Triple aus einem Subjekt, einem Prädikat und einem Objekt gebildet. Durch die Wiederverwendung einzelner Elemente in neuen Triples entsteht ein Informationsnetzwerk

Es gibt verschiedene Serialisierungen

Turtle [BB11] ist eine textuelle Serialisierung, die sehr gut von Menschen gelesen werden kann. Sie beinhaltet einige mächtige Abkürzungsmechanismen

RDF/XML [GSB14] ist eine der ersten Serialisierungen von RDF, die komplett auf XML basiert und somit von Anfang an die Möglichkeit bot, die Datensätze mit XML-Werkzeugen zu bearbeiten. Sie ist immer noch weit verbreitet, aber im Vergleich mit Turtle nur schwer von Menschen zu lesen.

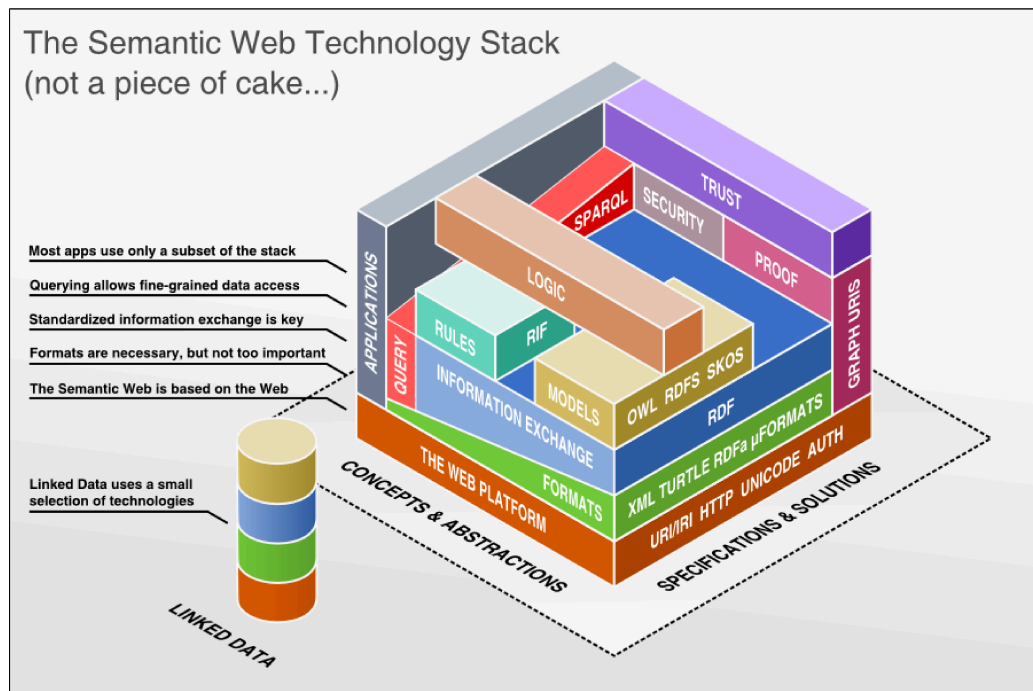


Abbildung C.1: Semantic Web Stack [zaz11b]

### C.1.3 HTTP

Das Hypertext Transfer Protocoll (HTTP) ist das am häufigsten genutzte Übertragungsprotokoll für das Internet. Es ist in RFC 2616 [Lea+99] spezifiziert und wird konsequenterweise auch zur Übertragung der Inhalte des Semantic Web genutzt. Von seinen Eigenschaften her ist HTTP ein zustandsloses Protokoll, dass auf der Anwendungsebene des ISO/OSI-Referenzmodells angesiedelt ist. Mit TLS (Transport Layer Security) bzw. dem Vorgänger SSL (Secure Sockets Layer) existiert eine Erweiterung von HTTP, das ein hybrides Verschlüsselungsprotokoll zur sicheren Datenübertragung beherrscht. somit wird aus dem Protokoll *http* die verschlüsselte Variante *https* (HyperText Transfer Protocol Secure) für die URIs der Ressourcen und Zugriffspunkte.

### C.1.4 SPARQL

SPARQL (SPARQL RDF Query Language) [PS08] ist die dominierende Abfragesprache für RDF und wird üblicherweise von allen TripleStores unterstützt. SPARQL nutzt eine graphenbasierte Sprache zur Beschreibung der Query und kommuniziert über HTTP.

### C.1.5 Links

Verknüpfungen (Links) zwischen verschiedenen Entitäten sind Bestandteile erster Ordnung in Linked Data. Das betrifft nicht nur Verknüpfungen zwischen Informationsentitäten innerhalb eines Datensatzes oder Servers. Durch die Nutzung von HTTP URIs kann eindeutig auf einer fremden Organisation zugehörige Informationseinheiten verwiesen werden.

### C.1.6 Ontologien

Die Semantic der verwendeten Konzepte und Relationen wird im Semantic Web durch Ontologien ausgedrückt. Eine Ontologie ist dabei nach [SBF98] “eine formale, explizite Spezifikation einer gemeinsamen Konzeptionalisierung”<sup>1</sup>.

Dabei sind *RDF Schema (RDFS)* und *Web Ontology Language (OWL)* die beiden Basis-Ontologien, mit deren Hilfe andere Ontologien erstellt werden. Dazu definieren diese Basiskonzepte wie Klassen, Properties und Vererbung. [BG04]

---

<sup>1</sup>Übersetzung des Autors von “formal, explicit specification of a shared conceptualisation.”

Die *Web Ontology Language (OWL)* [MH04] wurde durch OWL2 [W3C09] um Sets von Konzepten erweitert, die je nach Ausbaustufe Logiksysteme unterschiedlicher Mächtigkeit erlauben.

## C.2 Werkzeuge

### C.2.1 Frameworks

Für Linked Data gibt es eine ganze Reihe von Frameworks, die die Entwicklung von Applikationen erleichtern sowie auch das Bearbeiten von Linked Data ermöglichen. Diese gibt es für eine große Bandbreite von Programmiersprachen und Systemen (z.B. `rdflib`<sup>2</sup> für Python, `dotNetRDF`<sup>3</sup> für C# in .Net, `rdflquery`<sup>4</sup> für Javascript, ...), wobei sich im folgenden auf die populärsten zwei konzentriert wird.

**Sesame** Sesame<sup>5</sup> ist ein Open-Source Java Framework für die Speicherung und Abfrage von RDF Daten. Es ist in allen angebotenen Aspekten (Speichermechanismen, Inferenz-Engines, RDF Serialisierungen, Abfragesprachen und Abfrageergebnisformaten) einfach nutzbar und erweiterbar. Dabei ist es unter einer BSD-Lizenz freigegeben. Sesame bietet verschiedene APIs an sowie eine RESTful HTTP Schnittstelle, die das SPARQL Protokoll unterstützt.

**Jena** Jena<sup>6</sup> ist ein freies Java Framework für die Entwicklung von Anwendungen für das Semantic Web. Jena stellt unter der Apache Lizenz Version 2.0 eine Sammlung von Werkzeugen und Java Bibliotheken zur Verfügung, die beim Erstellen von Anwendungen, Werkzeugen und Servern für Linked Data unterstützen. Es bietet sowohl umfangreiche APIs also auch Kommandozeilenwerkzeuge für das Lesen, Verarbeiten und Schreiben von RDF Datensätzen sowie für SPARQL Abfragen, RDF Speicherung und Inferenz-Engines

### C.2.2 TripleStores

Es gibt eine Menge Arbeiten über einen Vergleich von Triplestores, z.B. [Nat09].

---

<sup>2</sup><https://github.com/RDFLib/rdflib>

<sup>3</sup><http://www.dotnetrdf.org/>

<sup>4</sup><https://code.google.com/p/rdflquery/>

<sup>5</sup><http://www.openrdf.org/>

<sup>6</sup><http://jena.apache.org/>

Dort werden unter anderem Funktionalität und Leistungsparameter ermittelt und miteinander verglichen.

Dabei spielen auch SPARQL Benchmarks eine große Rolle, wie z.B. der Berlin Benchmark<sup>7</sup> [].

In den letzten Jahren ist die Leistung der Triplestores enorm gestiegen, so dass sie nun ähnliche Größenordnungen wie relationalen Datenbanken erreichen [Erl14].

**Virtuoso** Virtuoso<sup>8</sup> ist eine industrielle Plattform zur Verwaltung von nativen Daten, Informationsmodellen und Wissen. Es unterstützt verschiedene Abfragesprachen, Protokolle, Datenrepräsentationen: SQL, SPARQL, ODBC, JDBC, HTTP, WebDAV, XML, RDF, RDFa, und einige weitere. Es existiert eine Open Source Variante mit dem Namen VOS (Virtuoso Open Source), die bereits einen hoch-performanten RDF Triplestore beinhaltet mit Unterstützung von Named Graphs.

Eine umfassende Dokumentation ist auf der Website<sup>9</sup> zu finden. Virtuoso steht einerseits in der offenen Variante unter der GPL Lizenz zur Verfügung, andererseits gibt es eine kommerzielle Version mit erweiterten Funktionen und Support.

**Stardog** Stardog<sup>10</sup> ist eine kommerzielle semantische Graphen-Datenbank. Sie unterstützt RDF als Datenmodell und SPARQL als Abfragesprache auf die mittels HTTP und SNARL Protokollen zugegriffen werden kann. Weiterhin kann über OWL 2 oder selbst definierte Regeln Inferenz- und Analysealgorithmen angewendet werden.

**Jena STB and TDB** Von dem freien und quelloffenen Framework Jena für das semantische Web gibt es ebenfalls zwei Triplestores. TDB<sup>11</sup> ist ein Triplestore, der auf Dateibasis arbeitet, während SDB<sup>12</sup> eine Implementierung darstellt, die direkt im Arbeitsspeicher läuft. Dabei ist TDB jedoch etwas schneller, stärker skalierbar und besser unterstützt als SDB.

---

<sup>7</sup><http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/>

<sup>8</sup><http://virtuoso.openlinksw.com/>

<sup>9</sup><http://ods.openlinksw.com/dataspace/dav/wiki/Main/VOS.VOSIndex>

<sup>10</sup><http://stardog.com/>

<sup>11</sup><https://jena.apache.org/documentation/tdb/index.html>

<sup>12</sup><https://jena.apache.org/documentation/sdb/>

### C.2.3 Ontologie-Management

**Protégé** Protégé<sup>13</sup> [Gen+03] ist ein freier, Open-Source Editor für die Erstellung und Verwaltung von Ontologien und Wissensbasen mit Exportmöglichkeiten in eine Reihe von Formaten (unter anderem RDF(S), OWL und XML Schema). Protégé nutzt ein erweiterbares Java-Framework und ermöglicht so die Erstellung und Nutzung von Plugins. Protégé richtet sich hauptsächlich an Ontologie-Experten und ist in der Nutzeroberfläche auf eine hohe Expertise des Nutzers ausgelegt. Protégé steht unter der Mozilla Public License (MPL) frei zur Verfügung.

**OntoSketch** OntoSketch<sup>14</sup> ist ein graphischer Editor für Mobilgeräte, der OWL Ontologien visualisieren kann und eine einfache Bearbeitung ermöglicht. Es ist dabei explizit für Nicht-Experten im Bereich des Wissensmanagement vorgesehen. Diese sollen damit befähigt werden bereits existierende Informationsmodelle auf ihre Bedürfnisse anzupassen um somit z.B. die Grundlage für eine Informationspartnerschaft zu schaffen. Aus diesem Grund sind nicht alle Sprachelemente von OWL unterstützt, sondern nur die grundlegenden Konzepte, die notwendig sind um Ontologien auf neue Anforderungen anzupassen.

Dazu gehören die Erstellung von Klassen, Properties und Individuals sowie das Zuweisen von Individuals zu einer Klasse und die Erstellung von Klassenhierarchien.

Für eine möglichst einfache Interaktion wird sich an die Konzepte vom händischen Skizzieren eines Informationsmodells auf Papier angelehnt. Dies wird durch die Nutzung eines Touchscreens und einer Symbol- und Handschrifterkennung bei einer Stifteingabe zur Benennung von Konzepten erreicht.

### C.2.4 Werkzeuge zur Verknüpfung von Datensätzen

**Silk** Silk<sup>15</sup> ist ein Werkzeug für das Entdecken von Verknüpfungen zwischen Entitäten aus verschiedenen Linked Data Datensätzen. Es ist ein quell-offenes Softwareprojekt, das auf JVMs läuft und somit auf allen verbreiteten Plattformen ausführbar ist. Flexible Verknüpfungsregeln werden in einer deklarativen Sprache hinterlegt, die mit einem graphischen Editor bearbeitet werden können. Erkannte Verknüpfungen können durch spezifizierte RDF

---

<sup>13</sup><http://protege.stanford.edu/>

<sup>14</sup><https://github.com/comvantage/OntoSketch>

<sup>15</sup><http://www4.wiwiss.fu-berlin.de/bizer/silk/>



Links miteinander verknüpft werden. Als Datenquellen und -senken können RDF Datensätze und SPARQL-Endpoints angebunden werden [Vol+09]

**LIMES** LIMES<sup>16</sup> ist ein weiteres Link-Entdeckungswerkzeug für das Semantic Web. Es sucht mit Hilfe von deklarativen Regeln bezüglich der Charakteristik von metrischen Räumen von Knoten nach ähnlichen Knoten. Die Regeln können über ein Web-Interface spezifiziert und die Ausführung gesteuert werden. Darüberhinaus bietet es Mechanismen für den Vorschlag von Links zwischen gefundenen Entitäten an.

---

<sup>16</sup><http://aksw.org/Projects/LIMES>



## Literaturverzeichnis

- [AB06] Rémi Arnaud und Mark C. Barnes. *COLLADA / sailing the gulf of 3D digital content creation*. A K Peters, 2006. xv+237. ISBN: 1-56881-287-6 (siehe S. 39).
- [AG212] AG2 M2M Initiative Deutschland. *Machine-to-Machine-Kommunikation - eine Chance für die deutsche Industrie*. Nov. 2012 (siehe S. 25–27).
- [Bar+94] William Barnett, Adrien Presley, Mary Johnson und D. H. Liles. „An architecture for the virtual enterprise“. In: *Systems, Man, and Cybernetics, 1994. Humans, Information and Technology, 1994 IEEE International Conference on*. Bd. 1. IEEE, 1994, S. 506–511. URL: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=399890](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=399890) (besucht am 13.10.2016) (siehe S. 10).
- [BB11] David Beckett und Tim Berners-Lee. *Turtle - Terse RDF Triple Language*. W3C Team Submission. 28. März 2011. URL: <http://www.w3.org/TeamSubmission/turtle/> (siehe S. 45, 125).
- [Ben99] David Benyon. „Social navigation of information space“. In: (1999). URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.95.9539> (besucht am 16.03.2015) (siehe S. 11).
- [Ber06] Tim Berners-Lee. *Design Issues - Linked Data*. 2006. URL: <http://www.w3.org/DesignIssues/LinkedData.html> (siehe S. 35).
- [Ber09] Mike Bergmann. *The Open World Assumption: Elephant in the Room / AI3:::Adaptive Information*. 21. Dez. 2009. URL: <http://www.mkbergman.com/852/the-open-world-assumption-elephant-in-the-room/> (besucht am 06.02.2015) (siehe S. 34).

- [BFM05] Tim Berners-Lee, Roy T. Fielding und Larry Masinter. *RFC 3986: Uniform Resource Identifier (URI): Generic Syntax*. Jan. 2005. URL: <https://tools.ietf.org/html/rfc3986> (besucht am 23.06.2014) (siehe S. 72, 125).
- [BG04] Dan Brickley und R. V. Guha. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Recommendation. 10. Feb. 2004. URL: <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/> (siehe S. 127).
- [BHB09] Christian Bizer, Tom Heath und Tim Berners-Lee. „Linked Data - The Story So Far“. In: *International Journal on Semantic Web and Information Systems (IJSWIS)* 5.3 (2009). Hrsg. von Tom Heath, M. Hepp und Christian Bizer, S. 1–22 (siehe S. 33).
- [Biz09] Christian Bizer. „The Emerging Web of Linked Data“. In: *Intelligent Systems, IEEE* 24.5 (2009), S. 87–92 (siehe S. 33).
- [BO13] Tim Berners-Lee und K. O’Hara. „The read-write Linked Data Web“. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1987 (18. Feb. 2013), S. 20120513–20120513. ISSN: 1364-503X, 1471-2962. DOI: 10.1098/rsta.2012.0513. URL: <http://rsta.royalsocietypublishing.org/cgi/doi/10.1098/rsta.2012.0513> (besucht am 25.10.2013) (siehe S. 46).
- [Boi13] Max Boisot. *Information space*. Routledge, 2013. URL: <https://books.google.de/books?hl=de&lr=&id=0a4wwRD6Sf8C&oi=fnd&pg=PP1&dq=%22information+space%22&ots=xdxemBb0-Q&sig=-SlPzOPXPE6-W8ccDlpJs2jNXG4> (besucht am 16.03.2015) (siehe S. 11).
- [Bra+12] Steven Braun, Christian Bartelt, Martin Obermeier, Andreas Rausch und Birgit Vogel-Heuser. „Requirements on evolution management of product lines in automation engineering“. In: *Int. Conf. Math. Modelling*. 2012, S. 340–345. URL: [http://seth.asc.tuwien.ac.at/proc12/full\\_paper/Contribution162.pdf](http://seth.asc.tuwien.ac.at/proc12/full_paper/Contribution162.pdf) (besucht am 03.07.2014) (siehe S. 29).
- [CH06] Krzysztof Czarnecki und Simon Helsen. „Feature-based survey of model transformation approaches“. In: *IBM Systems Jour-*

- nal* 45.3 (2006), S. 621–645. URL: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5386627](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5386627) (besucht am 27.02.2013) (siehe S. 16, 17).
- [CJS98] Mourad Chouikha, Axel Janhsen und Eckehard Schnieder. „Klassifikation und Bewertung von Beschreibungsmitteln für die Automatisierungstechnik“. In: *at - Automatisierungstechnik* 46.12 (1998), S. 582–591. DOI: 10.1524/auto.1998.46.12.582. URL: <https://www.degruyter.com/view/j/auto.1998.46.issue-12/auto.1998.46.12.582/auto.1998.46.12.582.xml> (besucht am 14.10.2016) (siehe S. 22).
- [Däu05] Lorenz Däubler. „Strukturverträgliche Ontologien der Automatisierungstechnik“. Diss. Braunschweig: TU Braunschweig, Nov. 2005. (Besucht am 04.06.2015) (siehe S. 14).
- [DB12] Rainer Drath und Mike Barth. „Semantikvielfalt mit AutomationML beherrschen“. In: *atp-edition* 54.12 (2012), S. 73–81 (siehe S. 43, 66).
- [DC12] DIN (Deutsches Institut für Normung) und CEN (Europäische Komitee für Normung). *DIN EN 62714-1: Datenaustauschformat für Planungsdaten industrieller Automatisierungssysteme (AutomationML) - Teil 1: Architektur und allgemeine Festlegungen*. Juni 2012 (siehe S. 39).
- [DCI09] DIN (Deutsches Institut für Normung), CEN (Europäische Komitee für Normung) und ISO (International Organization for Standardization). *DIN EN ISO 11354-1: Fortgeschrittene Automatisierungstechnologien und deren Anwendung*. Entwurf. Juni 2009 (siehe S. 5).
- [DD15] DIN (Deutsches Institut für Normung) und DKE (Deutsche Kommission Elektrotechnik). *Deutsche Normungsroadmap - Industrie 4.0 - Version 2*. 2015 (siehe S. 3, 27, 98).
- [DF04a] Rainer Drath und M. Fedai. „CAEX – ein neutrales Datenaustauschformat für Anlagendaten – Teil 1“. In: *atp edition – Automatisierungstechnische Praxis* (2004) (siehe S. 39).

- [DF04b] Rainer Drath und M. Fedai. „CAEX – ein neutrales Datenaustauschformat für Anlagendaten – Teil 2“. In: *atp edition – Automatisierungstechnische Praxis* (2004) (siehe S. 39).
- [DFB11] Rainer Drath, Alexander Fay und Mike Barth. „Interoperabilität von Engineering-Werkzeugen“. In: *at-Automatisierungstechnik* 59.7 (2011), S. 451–460 (siehe S. 10).
- [DI04] DIN (Deutsches Institut für Normung) und ISO (International Organization for Standardization). *DIN ISO 10303-212: Industrielle Automatisierungssysteme und Integration – Produktdarstellung und -austausch – Teil 212: Anwendungsprotokoll: Elektrische/elektrotechnische Systeme und Anlagen*. Dez. 2004 (siehe S. 49).
- [Die+14] Christian Diedrich, Matthias Meyer, Lars Evertz und Wilhelm Schäfer. „Dienste in der Automatisierungstechnik“. In: *atp edition – Automatisierungstechnische Praxis* 56.12 (1. Dez. 2014), S. 24. ISSN: 2364-3137, 2190-4111. DOI: 10.17560/atp.v56i12.467. URL: [http://ojs.di-verlag.de/index.php/atp\\_edition/article/view/467](http://ojs.di-verlag.de/index.php/atp_edition/article/view/467) (besucht am 13.04.2016) (siehe S. 28).
- [Die12] Reinhard Diestel. *Graph Theory*. GTM 173. Heidelberg: Springer, 2012. 451 S. URL: <http://diestel-graph-theory.com/> (siehe S. 16).
- [Dra10] Rainer Drath. *Datenaustausch in der Anlagenplanung mit AutomationML: Integration von CAEX, PLCopen XML und COLLADA*. Springer DE, 2010. URL: <http://books.google.com/books?hl=en&lr=&id=o5XfzefXSyAC&oi=fnd&pg=PA1&dq=%22B%C3%BCcher+behandeln+im+Allgemeinen+technische%22+%22Modell+kann+gewerke-%C3%BCbergreifend+erstellt+und+genutzt+werden.%22+%22vom+sp%C3%A4teren+Anlagenverhalten+verschaffen%22+%22innerhalb+erstaunlich+kurzer+Zeit+zu+beachtlicher+Reife+entwickelt+werden.%22+&ots=9h5tk-g34W&sig=Ao6phNU3vEQcPpvH0Zi4WBiYBWA> (besucht am 30.10.2013) (siehe S. 39).

- [DS05] M. Duerst und Michel Suignard. *RFC 3987: Internationalized Resource Identifiers (IRIs)*. Jan. 2005. URL: <http://tools.ietf.org/html/rfc3987> (besucht am 29.08.2014) (siehe S. 125).
- [Ehr+06] Hartmut Ehrig, Karsten Ehrig, Ulrike Prange und Gabriele Taentzer. *Fundamentals of algebraic graph transformation*. Berlin [etc.]: Springer, 2006. ISBN: 3-540-31187-4 978-3-540-31187-4 (siehe S. 61).
- [Ehr06] Marc Ehrig. *Ontology Alignment: Bridging the Semantic Gap: Bridging the Semantic Web Gap*. 1. Aufl. Springer US, 7. Nov. 2006. 248 S. ISBN: 0-387-32805-X (siehe S. 14).
- [Erl14] Orri Erling. „Virtuoso, The Prometheus of RDF - Semantics 2014 Conference Keynote“. Leipzig, 9. Apr. 2014. URL: <http://de.slideshare.net/kidehen/ogl-semantic-leipzigsemantics2014> (besucht am 28.07.2015) (siehe S. 129).
- [Fa13] Forschungsunion und acatech. *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0 - Abschlussbericht des Arbeitskreises Industrie 4.0*. Apr. 2013 (siehe S. 9, 22, 27).
- [Fia11] Fiatech. *An Introduction to ISO 15926*. Nov. 2011. URL: <https://www.posccaesar.org/wiki/ISO15926Primer> (siehe S. 36).
- [Fow95] Julian Fowler. *STEP for Data Management, Exchange and Sharing*. 1995. URL: <http://www.jpffowler.org/pdt/book/STEPbook.pdf> (besucht am 15.08.2016) (siehe S. 36).
- [FS05] Ernst Fricke und Armin P. Schulz. „Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle“. In: *Systems Engineering* 8.4 (2005). ISSN: 1098-1241, 1520-6858. DOI: 10.1002/sys.20039. URL: <http://doi.wiley.com/10.1002/sys.20039> (besucht am 03.06.2015) (siehe S. 24).
- [Gen+03] John H. Gennari, Mark A. Musen, Ray W. Ferguson, William E. Grosso, Monica Crubzy, Henrik Eriksson, Natalya F. Noy und Samson W. Tu. „The evolution of Protégé: an environment for knowledge-based systems development“. In: *International*

- Journal of Human-Computer Studies* 58.1 (2003), S. 89–123 (siehe S. 130).
- [GHU14] Markus Graube, Stephan Hensel und Leon Urbas. „R43ples: Revisions for Triples - An Approach for Version Control in the Semantic Web“. In: *Proceedings of the 1st Workshop on Linked Data Quality (LDQ)*. 1st Workshop on Linked Data Quality (LDQ). Hrsg. von Magnus Knuth, Dimitris Kontokostas und Harald Sack. Bd. 1215. CEUR Workshop Proceedings. 00007. Leipzig, 2014. URL: <http://ceur-ws.org/Vol-1215/#paper-03> (siehe S. 61, 85, 87, 88).
- [GHU16] Markus Graube, Stephan Hensel und Leon Urbas. „Open Semantic Revision Control with R43ples - Extending SPARQL to access revisions of Named Graphs“. In: SEMANTICS. Leipzig, 2016 (siehe S. 61, 85, 86).
- [GPP15] Sten Gruner, Julius Pfrommer und Florian Palm. „A RESTful extension of OPC UA“. In: *Factory Communication Systems (WFCS), 2015 IEEE World Conference on*. IEEE, 2015, S. 1–4. URL: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=7160557](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7160557) (besucht am 19.01.2016) (siehe S. 56).
- [Gra+11] Markus Graube, Johannes Pfeffer, Jens Ziegler und Leon Urbas. „Linked Data as Integrating Technology for Industrial Data“. In: *Proceedings of the 14th International Conference on Network-Based Information Systems*. 00038. IEEE, 7. Sep. 2011, S. 162–167. ISBN: 978-1-4577-0789-6. DOI: 10.1109/NBiS.2011.33. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6041917> (besucht am 04.11.2011) (siehe S. 49).
- [Gra+12] Markus Graube, Johannes Pfeffer, Jens Ziegler und Leon Urbas. „Linked Data as Integrating Technology for Industrial Data“. In: *International Journal of Distributed Systems and Technologies (IJDST)* 3.3 (2012). 00038, S. 40–52. URL: [doi:10.4018/jdst.2012070104](https://doi.org/10.4018/jdst.2012070104) (siehe S. 49).



- [Gra+13a] Markus Graube, Frank Haferkorn, Jan Hladik und Leon Urbas. „Bereitstellung von transienten Daten in Linked Data“. In: *Automation 2013*. Baden-Baden, 2013, S. 3–6 (siehe S. 71).
- [Gra+13b] Markus Graube, Patricia Ortiz, Manuel Carnerero, Oscar Lazaro, Mikel Uriarte und Leon Urbas. „Flexibility vs. Security in Linked Enterprise Data Access Control Graphs“. In: *Proc. of 9th IEEE Int. Conf. on Information Assurance and Security*. IAS. 00003. 2013, S. 13–18. DOI: <http://dx.doi.org/10.1109/ISIAS.2013.6947746> (siehe S. 66, 79).
- [Gra+13c] Markus Graube, Jens Ziegler, Jan Hladik und Leon Urbas. „Linked Data as Enabler for Mobile Applications for Complex Tasks in Industrial Settings“. In: *Proceedings of IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA 2013)*. 00004. IEEE, 2013, S. 1–8. (Besucht am 08.11.2011) (siehe S. 98).
- [Gra+14a] Markus Graube, Tobias Münch, Patricia Ortiz, Mohamed Hilla, Manuel Carnerero, Johannes Pfeffer, Jan Hladik, Philipp Herzig und Robert Buchmann. *D2.3 - ComVantage Architecture Best Practises*. Report (confidential) D2.3. ComVantage, 31. Aug. 2014 (siehe S. 70).
- [Gra+14b] Markus Graube, Patricia Ortiz, Manuel Carnerero, Oscar Lázaro, Mikel Uriarte, Johannes Pfeffer und Leon Urbas. „Flexibility vs. Security in Linked Enterprise Data Access Control Graphs“. In: *Journal of Information Assurance and Security* 9.2 (2014). 00003, S. 93–103. ISSN: 1554-1010. URL: <http://www.mirlabs.net/jias/secured/Volume9-Issue2/vol9-issue2.html> (siehe S. 79).
- [Gra09] Markus Graube. „Entwicklung einer netzwerkbasierter Datenschnittstelle zur Exploration von Wartungsaufträgen“. Studienarbeit. Dresden: Technische Universität Dresden, 2009 (siehe S. 78).
- [GSB14] Fabien Gandon, Guus Schreiber und David Beckett. *RDF 1.1 XML Syntax*. 25. Feb. 2014. URL: <http://www.w3.org/TR/rdf-syntax-grammar/> (siehe S. 125).

- [GU15] Markus Graube und Leon Urbas. „Modeling and Transformation of systems of systems using Linked Data“. In: *Proceedings of MATHMOD 2015*. MATHMOD 2015. 00000. Wien, 2015, S. 930–931. DOI: 10.3182/20150218-3-AU-30250//978-3-902823-71-70199 (siehe S. 64).
- [Hen13] Stephan Hensel. „Untersuchung von Synchronisierungsmechanismen in Linked Data Netzwerken“. Studienarbeit. Technische Universität Dresden, 2013 (siehe S. 85).
- [Hen14] Stephan Hensel. „Konflikterkennung und -behebung bei der Zusammenführung von reversionierten Graphen in Linked Data“. Diplomarbeit. Technische Universität Dresden, 2014 (siehe S. 86, 87).
- [HGU16] Stephan Hensel, Markus Graube und Leon Urbas. *Methodology for Conflict Detection and Resolution in Semantic Revision Control Systems*. PLT-Bericht 2016-08-A. Technische Universität Dresden - Professur für Prozessleittechnik, 2016 (siehe S. 87).
- [Hil15] Frank Hilbert. „Kontextabhängige Informationsräume - Unterstützung interdisziplinärer Bauinformationsprozesse durch eine kontextbewusste Informationslogik“. Diss. Dresden: Technische Universität Dresden, 28. Nov. 2015 (siehe S. 12).
- [HM85] Dennis Heimbigner und Dennis McLeod. „A Federated Architecture for Information Management“. In: *ACM Trans. Inf. Syst.* 3.3 (Juli 1985), S. 253–278. ISSN: 1046-8188. DOI: 10.1145/4229.4233. URL: <http://doi.acm.org/10.1145/4229.4233> (besucht am 16.03.2015) (siehe S. 12).
- [Hol+14] Thomas Holm, Michael Obst, Alexander Fay, Leon Urbas, Thomas Albers, Sven Kreft und Ulrich Hempfen. „Dezentrale Intelligenz für modulare Automation“. In: *atp edition - Automatisierungstechnische Praxis* 56.11 (1. Nov. 2014), S. 34. ISSN: 2364-3137, 2190-4111. DOI: 10.17560/atp.v56i11.353. URL: [http://ojs.di-verlag.de/index.php/atp\\_edition/article/view/353](http://ojs.di-verlag.de/index.php/atp_edition/article/view/353) (besucht am 13.04.2016) (siehe S. 28).

- [HP14] Patrick J. Hayes und Peter F. Patel-Schneider. *RDF 1.1 Semantics*. W3C Recommendation. 25. Feb. 2014. URL: <http://www.w3.org/TR/rdf11-mt/> (siehe S. 79).
- [IEC] IEC (International Electrotechnical Commission). *IEC 62541: OPC Unified Architecture* (siehe S. 37).
- [IEC16] IEC (International Electrotechnical Commission). *IEC 60050: International Electrotechnical Vocabulary*. 2016. URL: <http://www.electropedia.org/> (siehe S. 67).
- [II94] ISO (International Organization for Standardization) und IEC (International Electrotechnical Commission). *ISO IEC 7498-1: Informationstechnik - Kommunikation Offener Systeme - Basis-Referenzmodell*. Nov. 1994. URL: <http://www.beuth.de/de/norm/iso-iec-7498-1/1550939> (besucht am 25.02.2015) (siehe S. 47).
- [ISO07] ISO (International Organization for Standardization). *ISO 15926: Industrial automation systems and integration: Integration of life-cycle data for process plants including oil and gas production facilities*. 2007 (siehe S. 36, 49).
- [iX14] iX. *Linked Open Data erfreuen sich wachsender Beliebtheit*. iX. 25. Sep. 2014. URL: <http://www.heise.de/ix/meldung/Linked-Open-Data-erfreuen-sich-wachsender-Beliebtheit-2403748.html> (besucht am 29.07.2015) (siehe S. 33).
- [Jun14] Duan Junqi. „Weiterentwicklung eines Linked Data Adapters für OPC UA“. Studienarbeit. Technische Universität Dresden, 2014 (siehe S. 71).
- [Kec11] Christoph Kecher. *UML 2 / das umfassende Handbuch*. 4., aktualisierte und erw. Aufl. Galileo Press, 2011. 448 S. ISBN: 978-3-8362-1752-1 (siehe S. 90).
- [Kir07] Michael Kirmas. „Implementierung von Typicals“. In: *atp* 49.1 (2007), S. 18–19 (siehe S. 2).
- [Kön05] Alexander Königs. „Model transformation with triple graph grammars“. In: *Model Transformations in Practice Satellite*

- Workshop of MODELS*. 2005. URL: [http://sosym.dcs.kcl.ac.uk/events/mtip05/submissions/konigs\\_\\_model\\_transformation\\_with\\_triple\\_graph\\_grammars.pdf](http://sosym.dcs.kcl.ac.uk/events/mtip05/submissions/konigs__model_transformation_with_triple_graph_grammars.pdf) (besucht am 27.02.2013) (siehe S. 89).
- [Kön12] Barbara König. „Modellierung in der Informatik“. Vorlesung. Universität Duisburg Essen, 2012 (siehe S. 15).
- [KS06] Alexander Königs und Andy Schürr. „Tool Integration with Triple Graph Grammars - A Survey“. In: *Electronic Notes in Theoretical Computer Science* 148.1 (Feb. 2006), S. 113–150. ISSN: 15710661. DOI: 10.1016/j.entcs.2005.12.015. URL: <http://linkinghub.elsevier.com/retrieve/pii/S1571066106000454> (besucht am 28.02.2013) (siehe S. 18, 116).
- [KW07] E. Kindler und R. Wagner. „Triple graph grammars: Concepts, extensions, implementations, and application scenarios“. In: *University of Paderborn* (2007). URL: <http://svn-st.inf.tu-dresden.de/svn/reuseware/tags/20100220-emftext-1.2.2/Tornado/de.tudresden.tornado2.metamodel.modelC orrespondence/documentation/TGG-TechnicalReoprt.pdf> (besucht am 14.11.2012) (siehe S. 18, 90).
- [Lea+99] Paul J. Leach, Tim Berners-Lee, Jeffrey C. Mogul, Larry Masinter, Roy T. Fielding und James Gettys. *RFC 2616: Hypertext Transfer Protocol – HTTP/1.1*. Juni 1999. URL: <http://tools.ietf.org/html/rfc2616> (besucht am 29.06.2014) (siehe S. 127).
- [Lee99] Y. T. Lee. „Information modeling: From design to implementation“. In: *Proceedings of the second world manufacturing congress*. 1999, S. 315–321. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.5829&rep=rep1&type=pdf> (besucht am 03.02.2013) (siehe S. 15).
- [LG13] Rudolf Lauber und Peter Göhner. *Prozessautomatisierung 1: Automatisierungssysteme und -strukturen, Computer- und Busysteme für die Anlagen- und Produktautomatisierung, Echtzeitprogrammierung und Echtzeitbetriebssysteme, Zuverlässigkeits-*

- und Sicherheitstechnik. Springer-Verlag, 7. März 2013. 440 S. ISBN: 978-3-642-58446-6 (siehe S. 27).
- [Li12] Peng Li. „Einbindung dynamischer Daten in ein Linked Data Netzwerk von einem OPC-Server“. Studienarbeit. Technische Universität Dresden, 2012 (siehe S. 71).
- [Li13] Peng Li. „Integration dynamischer und historischer Prozessdaten in Linked Data Infrastrukturen“. Diplomarbeit. Technische Universität Dresden, 2013 (siehe S. 71).
- [Los+11] M. Loskyll, J. Schlick, S. Hodek, L. Ollinger, T. Gerber und B. Pîrvu. „Semantic service discovery and orchestration for manufacturing processes“. In: *2011 IEEE 16th Conference on Emerging Technologies Factory Automation (ETFA)*. 2011 IEEE 16th Conference on Emerging Technologies Factory Automation (ETFA). Sep. 2011, S. 1–8. DOI: 10.1109/ETFA.2011.6058988 (siehe S. 3).
- [LPH12] Danh Le-Phuoc, Josiane Xavier Parreira und Manfred Hauswirth. *Linked stream data processing*. Springer, 2012. URL: [http://link.springer.com/chapter/10.1007/978-3-642-33158-9\\_7](http://link.springer.com/chapter/10.1007/978-3-642-33158-9_7) (besucht am 22.01.2015) (siehe S. 55).
- [Mah+11] Wolfgang Mahnke, Andreas Gössling, Markus Graube und Leon Urbas. „Information modeling for middleware in automation“. In: *Proceedings of IEEE 16th Conference on Emerging Technologies & Factory Automation (ETFA 2011)*. 00008. IEEE, 2011, S. 1–7. ISBN: 978-1-4577-0017-0. DOI: 10.1109/ETFA.2011.6059111. URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6059111> (besucht am 08.11.2011) (siehe S. 22, 23, 37).
- [MH04] Deborah L. McGuinness und Frank van Harmelen. *OWL Web Ontology Language*. W3C Recommendation. 10. Feb. 2004. URL: <http://www.w3.org/TR/owl-features/> (siehe S. 128).
- [MLD09] Wolfgang Mahnke, Stefan-Helmut Leitner und Matthias Damm. *OPC Unified Architecture*. Dordrecht: Springer Berlin Heidelberg, 2009. ISBN: 978-3-540-68899-0 (siehe S. 37).

- [Mün13] Tobias Münch. *D2.2.2 - ComVantage Architecture Specification*. Report (confidential) D2.2.2. ComVantage, 30. Aug. 2013 (siehe S. 70).
- [NAM14] NAMUR. *NE 150: Standardisierte NAMUR-Schnittstelle zum Austausch von Engineering-Daten zwischen CAE-System und PCS-Engineering-Werkzeugen*. 13. Okt. 2014 (siehe S. 22, 24, 27, 29, 30).
- [Nat09] National Center for Biomedical Ontology. *Comparison of Triple Stores*. Feb. 2009. URL: [http://www.bioontology.org/wiki/index.php/RDF\\_Triple\\_Stores](http://www.bioontology.org/wiki/index.php/RDF_Triple_Stores) (siehe S. 128).
- [New96] Gregory B. Newby. „Metric Multidimensional Information Space“. In: *TREC* (1996) (siehe S. 11).
- [Obs+12] Michael Obst, Oliver Drumm, Falk Doherr, Christian Bauer und Leon Urbas. „Integriertes HMI-Engineering“. In: *Tagungsband Automation 2012*. Automation. 2012, S. 227–230 (siehe S. 90).
- [Obs+15a] Michael Obst, Thomas Holm, Leon Urbas, Alexander Fay, Sven Kreft, Ulrich Hempen und Thomas Albers. „Beschreibung von Prozessmodulen“. In: *atp edition - Automatisierungstechnische Praxis* 57.1 (1. Jan. 2015), S. 48. ISSN: 2364-3137, 2190-4111. DOI: 10.17560/atp.v57i01-02.473. URL: [http://ojs.dl-verlag.de/index.php/atp\\_edition/article/view/473](http://ojs.dl-verlag.de/index.php/atp_edition/article/view/473) (besucht am 13.04.2016) (siehe S. 28).
- [Obs+15b] Michael Obst, Thomas Holm, Leon Urbas, Alexander Fay, Sven Kreft, Ulrich Hempen und Thomas Albers. „Semantic description of process modules - Towards an open implementation for plug and produce in process plants“. In: ETFA. 2015. ISBN: 978-1-4673-7929-8 (siehe S. 28).
- [Ort+13] Patricia Ortiz, Oscar Lazaro, Mikel Uriarte und Manuel Carnero. „Enhanced Multi-Domain Access-Control for Secure Mobile Collaboration Through Linked Data Cloud In Manufacturing“. In: *Proceedings of IEEE World of Wireless Mobile and Multimedia Networks (WoWMoM) conference 2013*. IEEE WoWMoM 2013. 2013, S. 1–9 (siehe S. 59, 79).

- [PDM11] Kevin R. Page, David C. De Roure und Kirk Martinez. „REST and Linked Data: a match made for domain driven development?“ In: *Proceedings of the Second International Workshop on RESTful Design*. WS-REST '11. New York, NY, USA: ACM, 2011, S. 22–25. ISBN: 978-1-4503-0623-2. DOI: 10.1145/1967428.1967435. URL: <http://doi.acm.org/10.1145/1967428.1967435> (besucht am 08.02.2013) (siehe S. 55).
- [Pfe+12] Johannes Pfeffer, Markus Graube, Jens Ziegler und Leon Urbas. „Browsing Reversible Neighborhood Relations in Linked Data on Mobile Devices“. In: *Proceedings of the 2nd International Conference on Pervasive Embedded Computing and Communication Systems (PECCS 2012)*. 2nd International Conference on Pervasive Embedded Computing and Communication Systems (PECCS 2012). 00004. 2012, S. 150–155. DOI: <http://dx.doi.org/10.5220/0003823901500155> (siehe S. 98).
- [Pfe+13] Johannes Pfeffer, Markus Graube, Jens Ziegler und Leon Urbas. „Vernetzte Apps für komplexe Aufgaben in der Industrie“. In: *atp edition* 55.3 (1. März 2013). 00000, S. 34–41. ISSN: 2190-4111. URL: <https://www.di-verlag.de/de/Zeitschriften/atp-edition/2013/03/Vernetzte-Apps-fuer-komplexe-Aufgaben-in-der-Industrie> (besucht am 12.11.2013) (siehe S. 70, 98).
- [Pla15] Plattform Industrie 4.0. *Umsetzungsstrategie Industrie 4.0 - Ergebnisbericht der Plattform Industrie 4.0*. Apr. 2015 (siehe S. 9, 22, 27, 29).
- [Pri14] PriceWaterhouseCoopers. *Industrie 4.0 - Chancen und Herausforderungen der vierten industriellen Revolution*. 2014 (siehe S. 22).
- [PS08] Eric Prud'hommeaux und Andy Seaborne. *SPARQL Query Language for RDF*. W3C Recommendation. 15. Jan. 2008. URL: <http://www.w3.org/TR/rdf-sparql-query/> (siehe S. 127).
- [PU15] Johannes Pfeffer und Leon Urbas. „Towards formal modeling of App-Ensembles“. In: (2015). DOI: 10.13140/RG.2.1.2024.2322.

- URL: <http://dx.doi.org/10.13140/RG.2.1.2024.2322> (besucht am 09.08.2016) (siehe S. 70).
- [Rah16] Julian Rahm. „Roundtrip Engineering zwischen Detail-Engineering und HMI-Engineering in der Prozessindustrie“. Diplomarbeit. Dresden: Technische Universität Dresden, 2016 (siehe S. 61).
- [Rak+13] Nur Aini Rakhmawati, Jürgen Umbrich, Marcel Karnstedt, Ali Hasnain und Michael Hausenblas. „Querying over Federated SPARQL Endpoints—A State of the Art Survey“. In: *arXiv preprint arXiv:1306.1723* (2013). URL: <http://arxiv.org/abs/1306.1723> (besucht am 12.09.2014) (siehe S. 49, 52, 54).
- [SBF98] Rudi Studer, V. Richard Benjamins und Dieter Fensel. „Knowledge engineering: Principles and methods“. In: *Data & Knowledge Engineering* 25.1 (März 1998), S. 161–197. ISSN: 0169-023X. DOI: 10.1016/S0169-023X(97)00056-6. URL: <http://www.sciencedirect.com/science/article/pii/S0169023X97000566> (besucht am 01.02.2012) (siehe S. 15, 127).
- [SC09] J. F. Sequeda und O. Corcho. „Linked stream data: A position paper“. In: (2009). URL: [http://oa.upm.es/5442/1/INVE\\_MEM\\_2009\\_64353.pdf](http://oa.upm.es/5442/1/INVE_MEM_2009_64353.pdf) (besucht am 29.06.2012) (siehe S. 55).
- [Sch+14a] Nicole Schmidt, Arndt Lüder, Heinrich Steiniger und Stefan Biffl. „Analyzing Requirements on Software Tools According to the Functional Engineering Phase in the Technical Systems Engineering Process“. In: *Proc. of ETFA 2014*. IEEE ETFA 2014. Barcelona, 2014 (siehe S. 29).
- [Sch+14b] Guus Schreiber, Yves Raimond, Frank Manola, Eric Miller und Brian McBride. *RDF 1.1 Primer*. 25. Feb. 2014. URL: <http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225/> (siehe S. 125).
- [Sch07] Rainer Jürgen Schack. „Methodik zur bewertungsorientierten Skalierung der Digitalen Fabrik“. Diss. Technische Universität München, 2007. 260 S. (siehe S. 22, 25, 26, 30).
- [Sch13] Eckehard Schnieder. *Methoden der Automatisierung: Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisie-*



- runungssysteme*. Google-Books-ID: OFzPBgAAQBAJ. Springer-Verlag, 13. März 2013. 373 S. ISBN: 978-3-322-90879-7 (siehe S. 22, 96).
- [Sch94] Andy Schürr. *Specification of Graph Translators with Triple Graph Grammars*. Aachener Informatik-Bericht AIB 94-12. RWTH Aachen, 1994 (siehe S. 18).
- [SHS14] Sebastian Schlund, Moritz Hämmerle und Tobias Strölin. *Industrie 4.0 – Eine Revolution der Arbeitsgestaltung*. Hrsg. von Ingenics AG. 2014 (siehe S. 22, 23).
- [SMS11] Miriam Schleipen, Ansgar Münnemann und Olaf Sauer. „Interoperabilität von Manufacturing Execution Systems (MES)“. In: *at - Automatisierungstechnik* 59.7 (Juli 2011), S. 413–424. ISSN: 0178-2312. DOI: 10.1524/auto.2011.0936. URL: <http://www.oldenbourg-link.com/doi/abs/10.1524/auto.2011.0936> (besucht am 03.04.2013) (siehe S. 1, 21).
- [SN14] Muhammad Saleem und Axel-Cyrille Ngonga Ngomo. „HiBIS-CuS: Hypergraph-Based Source Selection for SPARQL Endpoint Federation“. In: *The Semantic Web: Trends and Challenges*. Hrsg. von Valentina Presutti, Claudia d’Amato, Fabien Gandon, Mathieu d’Aquin, Steffen Staab und Anna Tordai. Bearb. von David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Alfred Kobsa, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Demetri Terzopoulos, Doug Tygar und Gerhard Weikum. Bd. 8465. Cham: Springer International Publishing, 2014, S. 176–191. ISBN: 978-3-319-07442-9 978-3-319-07443-6. URL: [http://link.springer.com/10.1007/978-3-319-07443-6\\_13](http://link.springer.com/10.1007/978-3-319-07443-6_13) (besucht am 29.09.2015) (siehe S. 54).
- [Spa+13] Dieter Spath, Oliver Ganschar, Stefan Gerlach, Moritz Hämmerle, Tobias Krause und Sebastian Schlund. *Produktionsarbeit der Zukunft-Industrie 4.0*. Hrsg. von Dieter Spath. Fraunhofer Verlag, 2013. URL: [http://www.dkp-niedersachsen.de/produktion/medien/archiv/20131109wesem/Fraunhofer-IA0-Studie\\_Produktionsarbeit\\_der\\_Zukunft-Industrie\\_4.0.pdf](http://www.dkp-niedersachsen.de/produktion/medien/archiv/20131109wesem/Fraunhofer-IA0-Studie_Produktionsarbeit_der_Zukunft-Industrie_4.0.pdf) (besucht am 16.05.2014) (siehe S. 10, 22, 24, 29, 62).

- [Sta73] Herbert Stachowiak. *Allgemeine Modelltheorie*. Springer-Verlag Wien New York, 1973. ISBN: 3-211-81106-0 (siehe S. 15).
- [Ste12] Peter M. Stephan. „Entwicklung einer Referenzarchitektur zur Nutzung semantisch interpretierter Ortsinformationen am Beispiel der Instandhaltung“. Diss. Technische Universität Kaiserslautern, 2012. (Besucht am 26.06.2012) (siehe S. 24).
- [Str+11] M. Strube, S. Runde, H. Figalist und A. Fay. „Risk minimization in modernization projects of plant automation #x2014; A knowledge-based approach by means of semantic web technologies“. In: *2011 IEEE 16th Conference on Emerging Technologies Factory Automation (ETFA)*. 2011 IEEE 16th Conference on Emerging Technologies Factory Automation (ETFA). Sep. 2011, S. 1–8. DOI: 10.1109/ETFA.2011.6058987 (siehe S. 3).
- [Sun14] Qingnan Sun. „Konzeption und Entwicklung einer Pilotanwendung für Industrie 4.0“. Masterarbeit. Technische Universität Dresden, 2014 (siehe S. 98).
- [SWH09] Heinrich Schmedding, Hans Wende und Jens Heinrich. „10000 Feldgeräte am Bus - ein Erfahrungsbericht aus China“. In: *atp* 86 (2009) (siehe S. 2).
- [SWZ95] Andy Schürr, Andreas J. Winter und Albert Zündorf. „Graph grammar engineering with PROGRES“. In: *Software Engineering — ESEC ’95*. Hrsg. von Wilhelm Schäfer und Pere Botella. Bearb. von G. Goos, J. Hartmanis und J. van Leeuwen. Bd. 989. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, S. 219–234. ISBN: 978-3-540-60406-8 978-3-540-45552-3. URL: [http://link.springer.com/10.1007/3-540-60406-5\\_17](http://link.springer.com/10.1007/3-540-60406-5_17) (besucht am 11.08.2016) (siehe S. 63).
- [Tae+05] Gabriele Taentzer, Karsten Ehrig, Esther Guerra, Juan De Lara, Laszlo Lengyel, Tihamer Levendovszky, Ulrike Prange, Dániel Varró und Szilvia Varró-Gyapay. „Model transformation by graph transformation: A comparative study“. In: *Proc. Workshop Model Transformation in Practice, Montego Bay, Jamaica*. 2005. URL: <http://www.mathematik.uni-marburg.de/~swt/Publ>

- ikationen\_Taentzer/TEG+05.pdf (besucht am 20.09.2013)  
(siehe S. 17).
- [The+09] S. Theurich, Ch. Hahn, R. Frenzel und M. Wollschlaeger. „Field bus abstraction as a means to enable network-independent applications“. In: *Proceedings of 8th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*. Bd. 8. Hanyang University, Republic of Korea, 2009, S. 131–138 (siehe S. 2, 3).
- [TS08] Andrew S. Tanenbaum und Maarten van Steen. *Verteilte Systeme / Prinzipien und Paradigmen*. 2., aktualisierte Aufl. Pearson Studium, 2008. 761 S. ISBN: 978-3-8273-7293-2 (siehe S. 13).
- [UPZ11] Leon Urbas, Johannes Pfeffer und Jens Ziegler. „iLD-Apps: Usable Mobile Access to Linked Data Clouds at the Shop Floor“. In: *Proc. Workshop Visual Interfaces to the Social and the Semantic Web (VISSW 2011)*. 2011 (siehe S. 98).
- [Van+13] Miel Vander Sande, Pieter Colpaert, Ruben Verborgh, Sam Coppens, Erik Mannens und Rik Van de Walle. „R&Wbase: Git for triples“. In: *Proceedings of the 6th Workshop on Linked Data on the Web*. 2013. URL: <http://events.linkedata.org/ldow2013/papers/ldow2013-paper-01.pdf> (besucht am 26.07.2013) (siehe S. 85).
- [VDI11a] VDI. *VDI 4499-2: Digitale Fabrik - Digitaler Fabrikbetrieb*. 2011 (siehe S. 22, 26).
- [VDI11b] VDI/VDE. *VDI/VDE 2657-1: Middleware in der Automatisierungstechnik - Grundlagen*. Sep. 2011 (siehe S. 10, 11, 13, 14, 23–26).
- [VDI13] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik. *Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation*. Apr. 2013 (siehe S. 9, 11, 22, 24, 25, 27).
- [VDI14] VDI/VDE GMA-Fachausschuss "Industrie 4.0". *Industrie 4.0 - Auf dem Weg zu einem Referenzmodell*. Apr. 2014 (siehe S. 25).

- [VDI15] VDI/VDE. *VDI/VDE 2657-2: Middleware in der Automatisierungstechnik - Vorgehensmodell für den Middleware-Engineering-Prozess*. 2015 (siehe S. 14).
- [Vol+09] Julius Volz, Christian Bizer, Martin Gaedke und Georgi Kobilarov. „Discovering and Maintaining Links on the Web of Data“. In: *The Semantic Web - ISWC 2009*. Hrsg. von Abraham Bernstein, David Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta und Krishnaprasad Thirunarayan. Bd. 5823. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, S. 650–665. URL: [http://dx.doi.org/10.1007/978-3-642-04930-9\\_41](http://dx.doi.org/10.1007/978-3-642-04930-9_41) (siehe S. 131).
- [W3C09] W3C OWL Working Group. *OWL 2 Web Ontology Language*. W3C Recommendation. 27. Okt. 2009. URL: <http://www.w3.org/TR/owl2-overview/> (siehe S. 128).
- [W3C11] W3C. *W3C Semantic Web Activity*. W3C Website. 7. Nov. 2011. URL: <http://www.w3.org/2001/sw/> (siehe S. 33).
- [W3C13] W3C. *PROV-O: The PROV Ontology*. W3C Recommendation. 20. Apr. 2013. URL: <http://www.w3.org/TR/prov-o/> (siehe S. 86).
- [Wie12] Andreas Wiesner. „An Ontology-based Framework for Design Data Consolidation in Chemical Process Engineering“. Diss. RWTH Aachen, 2012 (siehe S. 36).
- [Yan15] Xinyu Yang. „Erweiterte Merging-Funktionalitäten für semantische Revisionsverwaltungssysteme“. Studienarbeit. Technische Universität Dresden, 2015 (siehe S. 86, 87).
- [zaz11a] zazi0815. *A generalisation of the Linked Data publishing guideline*. Some More Individual. 17. Feb. 2011. URL: <http://smiy.wordpress.com/2011/02/17/a-generalisation-of-the-linked-data-publishing-guideline/> (besucht am 27.06.2014) (siehe S. 35).
- [zaz11b] zazi0815. *The common, layered Semantic Web technology stack*. Some More Individual. 1. Okt. 2011. URL: <http://smiy.wordpress.com/2011/01/10/the-common-layered-semantic->

- `web-technology-stack/` (besucht am 01.03.2012) (siehe S. 46, 126).
- [ZGU12] Jens Ziegler, Markus Graube und Leon Urbas. „RFID as universal entry point to linked data clouds“. In: *2012 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*. 2012 IEEE International Conference on RFID-Technologies and Applications (RFID-TA). 00007. 2012, S. 281–286. DOI: 10.1109/RFID-TA.2012.6404530 (siehe S. 98).
- [Zie+12] Jens Ziegler, Markus Graube, Johannes Pfeffer und Leon Urbas. „Beyond App-Chaining: Mobile App Orchestration for Efficient Model Driven Software Generation“. In: *17th international IEEE Conference on Emerging Technologies & Factory Automation. ETFA 2012*. 00017. Krakau, Poland, 2012, S. 1–8 (siehe S. 70, 98).
- [Zie+14] Jens Ziegler, Robert Buchmann, Markus Graube, Jan Hladik, Tobias Münch, Patricia Ortiz, Johannes Pfeffer, Florian Schneider, Mikel Uriarte, Dimitris Karagiannis und Leon Urbas. „Implementation and Operation of Collaborative Manufacturing Networks“. In: *Collaborative Systems for Smart Networked Environments*. Hrsg. von Luis M. Camarinha-Matos und Hamideh Afsarmanesh. Bd. 434. 00000. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 197–208. ISBN: 978-3-662-44744-4 978-3-662-44745-1. URL: [http://link.springer.com/10.1007/978-3-662-44745-1\\_19](http://link.springer.com/10.1007/978-3-662-44745-1_19) (besucht am 22.10.2014) (siehe S. 67, 68).